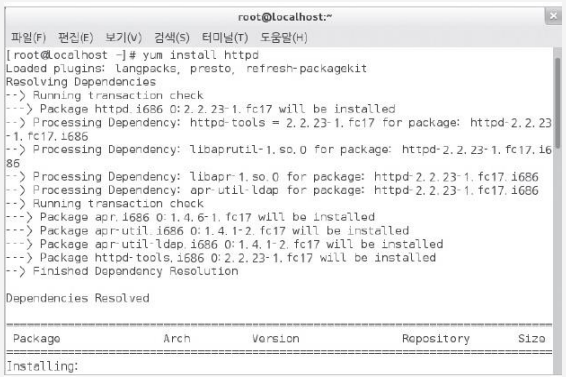
1 시스템 보안 요소

1 아파치 웹 서버 설치하기

1 httpd 설치

- 페도라에서 yum을 이용하여 아파치 설치
- yum install httpd



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 설치하기

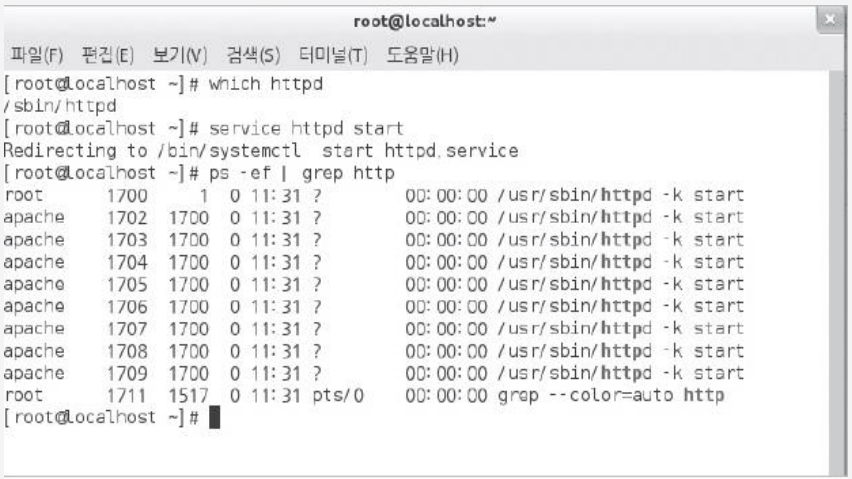
2 httpd 서비스 시작

- httpd 서비스를 구동하는 명령어(service)
  : ‘service httpd start’
- httpd 서비스가 정상적으로 구동되고 있는지 확인(ps, grep)
  : ‘ps -ef | grep http’

1 아파치 웹 서버 설치하기

2 httpd 서비스 시작

[페도라에서 httpd 구동]

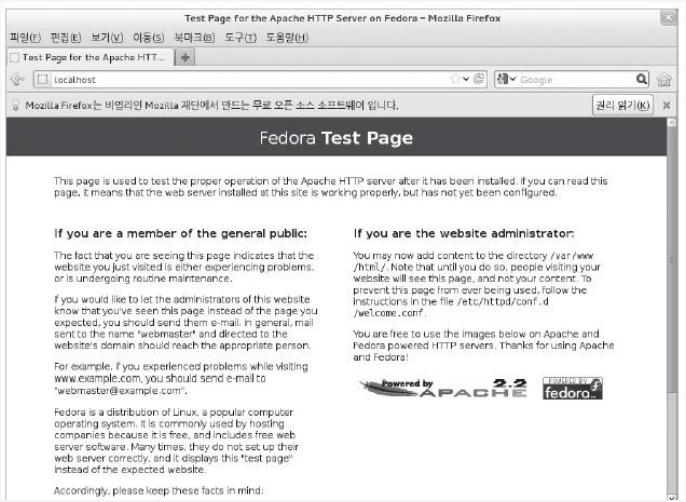


※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 설치하기

3 설치 결과 확인(localhost)

- 브라우저를 통해 웹 사이트를 접속해 설치 및 구동 확인



[페도라에서 아파치 설치 확인]

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

2 아파치 웹 서버

- 아파치 HTTP 서버(영어: Apache HTTP Server)는
  아파치 소프트웨어 재단에서 관리하는 HTTP 웹 서버임,

  BSD, 리눅스 등 유닉스 계열 뿐 아니라
  마이크로소프트 윈도우나 노벨 넷웨어 같은
  기종에서도 운용할 수 있음(No.1)

2 아파치 웹 서버

활용

- 리눅스 운영 체제, 아파치 웹 서버, MySQL 데이터베이스, PHP등으로 웹 서버를 운영하는 것을 각각의 머릿글자를 따서 LAMP라고도 부르기도 함
- 톰캣(Tomcat), Resin 등의 웹 애플리케이션 서버와 같이 사용할 수 있음
- Open-SSL, Mod-SSL 을 설치하여, 보안을 강화할 수 있음(http → https)

2 아파치 웹 서버

- 점유율
  - 아파치 웹 서버는 현재 세계에서 가장 인기 있는 웹 서버임
  - 2017년 10월 기준으로 실질적으로 작동하는 웹 사이트(Active site)들에서 쓰이는 웹 서버 소프트웨어 순위는 아파치(44.89%), 엔진엑스(20.65%), 구글 웹 서버(7.86%), 마이크로소프트 IIS(7.32%)순 임

2 아파치 웹 서버

- 점유율
  - 이 조사에서 생성은 되어있으나 정상적으로 작동하지 않는 웹 사이트들은 배제되었으며 특히 MS의 인터넷 정보 서비스(IIS)를 설치한 웹 사이트들의 상당수가 비활성 사이트였음, 그런 사이트들도 포함하면 MS IIS가 1위임
  - 2017년 3월 현재 Apache는 한국 전체 등록 도메인 중 42.39%가 사용하고 있음

2 아파치 웹 서버

- 리눅스버전 설치 예
  - 페도라에서 아파치 설치
    : yum install httpd
  - 데비안계열 우분투 메이트(ARM)에서 아파치 설치
    : apt-get install apache2

2 아파치 웹 서버

- 리눅스버전 설치 예
  - 아파치의 핵심 설정 파일은
    /etc/httpd/conf/httpd.conf
    또는 /etc/apache2/apache2.conf 임

    여기에 딸려있는 하위 파일 중에서
    SeverName 항목은 /etc/apache2/sites-enabled/000-default.conf에 있음, 아파치가 설치되면 localhost인 http://127.0.0.1에서 초기화면을 확인 할 수 있음

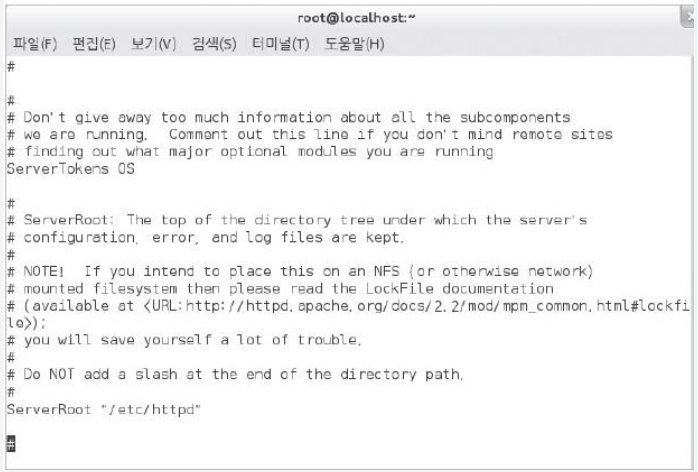
2 시스템 보안 설정

1 아파치 웹 서버 보안 설정하기

1 아파치 설치 디렉터리 확인(vi)

- vi /etc/httpd/conf/httpd.conf

[아파치 서버의 경로 설정하기]



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

1 아파치 설치 디렉터리 확인

- 중요한 파일에는 적절한 권한 설정이 필요함

[아파치의 주요 파일에 대한 접근 권한]

디렉터리	소유자	그룹	접근 권한
httpd	root	root	511
passwd	root	nobody	640
httpd.conf	root	nobody	640

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

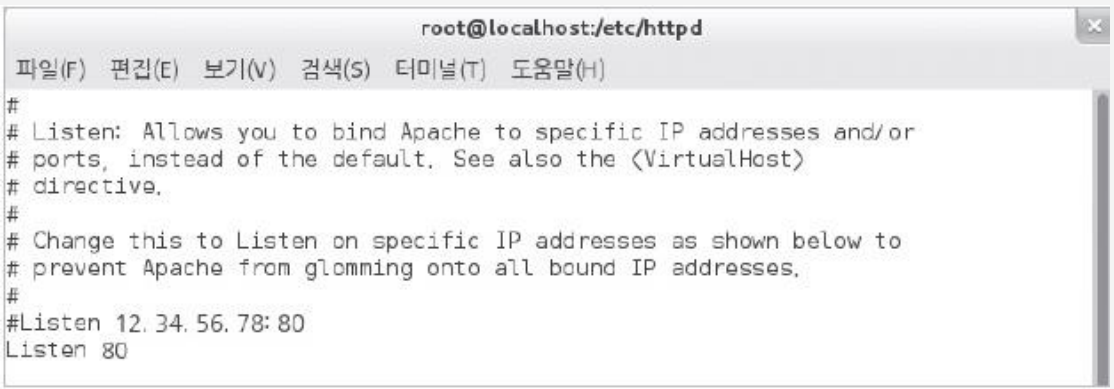
2 서버 이름과 IP, 포트 설정

- 기본 설정 : 서버의 80번 포트를 이용하여 접속
- 특정 인터페이스로만 서비스를 제공할 경우 Listen을 이용하여 설정

1 아파치 웹 서버 보안 설정하기

2 서버 이름과 IP, 포트 설정(IP, Port)

[httpd.conf로 IP 포트 설정]



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

2 서버 이름과 IP, 포트 설정(서버 이름)

[httpd.conf로 IP 포트 설정]

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

3 서버 실행 계정과 그룹 설정 확인(apache)

- ‘apache’라는 별도의 계정을 nobody 권한으로 만들어 아파치 웹 프로세스에 대한 권한으로 할당

root@localhost:/etc/httpd

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
#    suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group #-1 on these systems!
#
User apache
Group apache
```

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

3 서버 실행 계정과 그룹 설정 확인

- cat /etc/passwd

```
anesra:x:1000:1000:anesra:/home/anesra:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
[root@localhost httpd]# _
```

- /etc/passwd 파일에서 '/sbin/nologin'으로 나타나므로 로그인 권한이 없음(권한 상승)

```
anesra:x:1000:
apache:x:48:
[root@localhost httpd]#
```

- gid 값이 0으로 되어 있으면 root 그룹이라는 뜻이므로 apache가 일반 그룹을 확인 할 수 있음

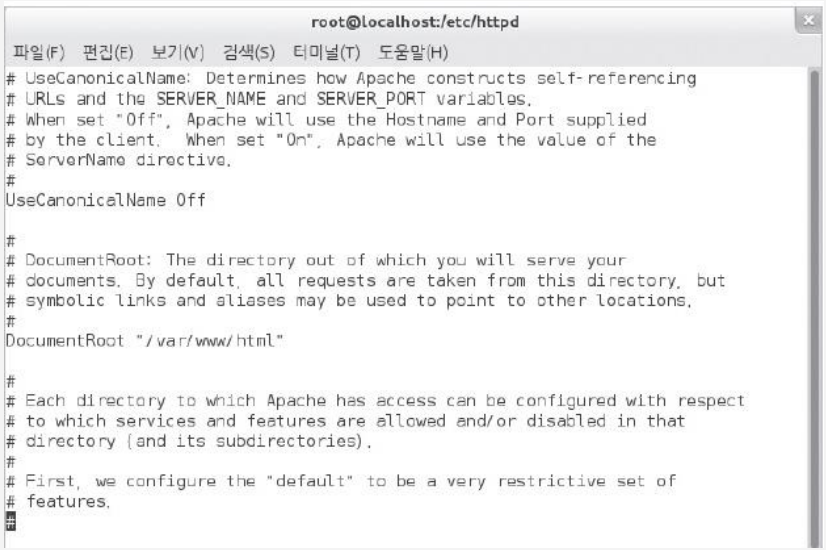
1 아파치 웹 서버 보안 설정하기

4 웹 페이지 디렉터리 설정 확인

- httpd.conf 파일에서는 웹 페이지 루트 경로를 설정할 수 있음(기본 웹 페이지 루트 경로는 /var/www/html로 설정되어 있음)
- DocumentRoot는 공격자가 추측하기 어려운 경로로 만들거나 중요한 디렉터리에는 접근할 수 없도록 제한된 하위 디렉터리로 지정해야 함 (중요 파일 접근)

1 아파치 웹 서버 보안 설정하기

4 웹 페이지 디렉터리 설정 확인(DocumentRoot)

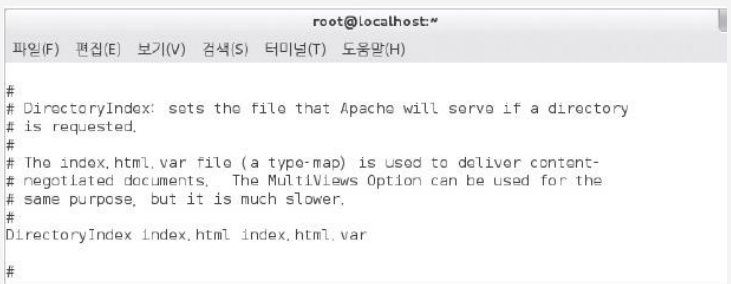


※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

5 기본 문서 설정 확인

- 기본 문서 설정을 확인하기 위해 httpd.conf 파일에서 DirecotryIndex 문자열을 찾음
- 기본 설정은 index.html이 가장 높은 우선순위로 되어 있음

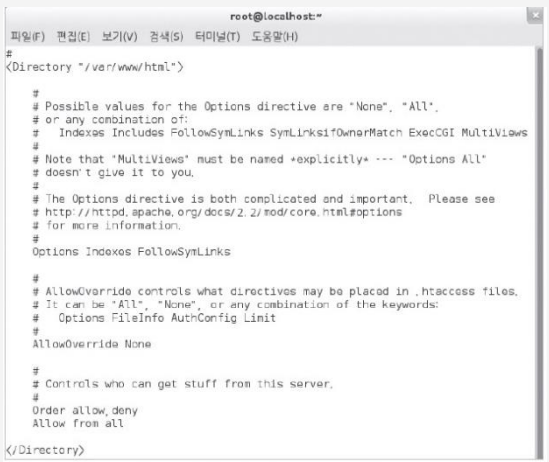


※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

6 디렉터리 리스팅 설정 확인(디렉터리 리스팅)

- 디렉터리의 Options에 Indexes가 설정되어 있다면 디렉터리 리스팅이 불가능하도록 삭제해야 함



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

6 디렉터리 리스팅 설정 확인(중요 파일 접근)

- 디렉터리 리스팅이 되어있으면
  해당 폴더의 하위 파일을 볼 수 있음

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

6 디렉터리 리스팅 설정 확인(Options → Indexes)

- 디렉터리 리스팅의 취약점을 제거하려면 httpd.conf 파일의 Options 부분에 있는 Indexes를 삭제해야 함

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

7 FollowSymLinks 설정 확인

- FollowSymLinks는 ServerRoot, DocumentRoot 설정과는 무관하게 중요 시스템 디렉터리로의 접근이 가능해지기 때문에 반드시 주의해야 함

1 아파치 웹 서버 보안 설정하기

7 FollowSymLinks 설정 확인(하드 vs. 소프트 링크)

- /etc 디렉터리에 대한 심벌릭 링크 생성(바로가기)
- ln -s /etc /var/www/html/etc

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

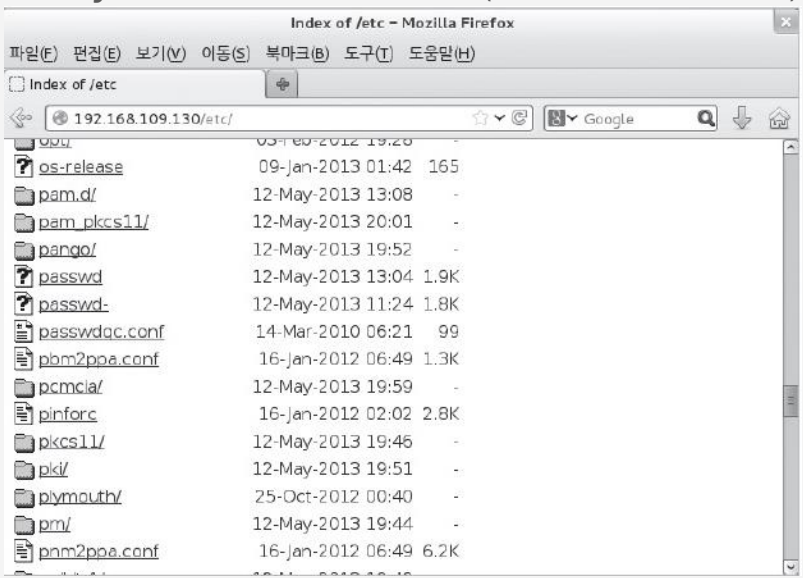
1 아파치 웹 서버 보안 설정하기

7 FollowSymLinks 설정 확인

- http://192.168.109.130/etc/에 접속하면 /etc 디렉터리의 여러 파일에 바로 접근할 수 있으며, passwd 파일도 읽을 수 있음
- 반드시 필요한 경우가 아니라면 Options에서 FollowSymLinks를 삭제해야 함

1 아파치 웹 서버 보안 설정하기

7 FollowSymLinks 설정 확인(중요 파일 접근)



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 아파치 웹 서버 보안 설정하기

8 접근 제어의 확인과 설정(Allow, Deny)

- DocumentRoot로 설정되어 있는 /var/www/html에 대한 접근 제어 확인

```
<Directory "/var/www/html">
    ⋮
  Order Allow, Deny
  Allow from all
    ⋮
</Directory>
```

1 아파치 웹 서버 보안 설정하기

8 접근 제어의 확인과 설정

- 특정 IP에 대한 거부 설정(Black list)

```
Order Allow, Deny
Allow from all
Deny from 11.22.33.44
```

- 특정 도메인에 대한 접근 허락 설정(White list)

```
Order Deny, Allow
Deny from all
Allow from .hacker.com
```

1 아파치 웹 서버 보안 설정하기

9 ServerSignature 설정

- 외부에서 아파치 서버의 버전과 운영체제의 종류를 알아내지 못하도록 할 수 있음 (다양한 정보 → 해킹)
- ServerSignature 값은 On, Off, Email로 설정(기본값은 On)

1 아파치 웹 서버 보안 설정하기

9 ServerSignature 설정

root@localhost:~

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature On
```

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

2 Httpd.conf - global environment

ServerRoot	서버의 루트를 결정한다.
LockFile	잠금 파일의 위치를 기록한다.
ScoreBoardFile	서버 프로세스의 상태를 기록하는 파일의 위치를 지정한다.
PidFile	서버가 동작할 때 생기는 프로세스의 PID를 기록하는 파일 위치를 지정한다.
Timeout	클라이언트가 서버로부터 파일을 받을 때 대기하는 최대 시간을 초단위로 지정한다.
KeepAlive On	자식 프로세스를 계속 유지하여 효율성을 증가시킨다.
MaxKeepAliveRequests	KeepAlive가 설정된 경우 몇 번의 연결 요청을 받을지 결정한다.
KeepAliveTimeout	KeepAlive가 설정된 경우 초 단위로 지정된 시간 후에 연결을 종료한다.
StartServers	몇 개의 프로세스로 시작할지 지정한다.
MinSpareServers, MaxSpareServers	서버 풀의 크기를 조절한다.
MaxClients	동시에 접속할 수 있는 클라이언트 최대 개수로, 최대 프로세스의 수를 지정한다.
MaxRequrestPerChild	하나의 자식 프로세스가 처리하는 최대 연결 요청 수를 설정한다.
Listen	아파치의 포트를 설정한다. 기본값은 80이다.

3 Httpd.conf - main server configuration

User와 Group	아파치 서버를 사용할 사용자와 그룹을 지정한다.
ServerAdmin	서버에 문제가 생겼을 때 연락할 메일 주소를 지정한다.
ServerName	서버 이름을 지정한다.
DocumentRoot	웹 문서의 위치를 지정한다.
<Directory/> </Directory>	각 디렉터리에 대한 접근 권한을 설정한다.
UserDir	사용자 홈페이지 디렉터리를 지정한다.
ErrorLog	에러 로그가 저장될 위치를 지정한다.
LogLevel	로그 레벨을 지정한다.
LogFormat	CustomLog에서 사용하는 로그 포맷 별칭을 정한다.
CustomLog	로그 파일 이름과 형식을 지정한다.

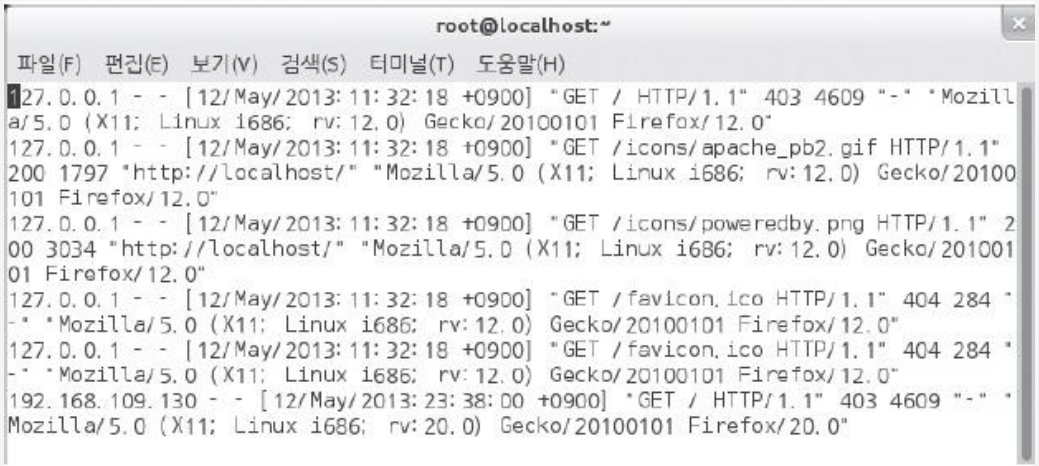
4 Httpd.conf - virtual hosts

NameVirtualHost	버츄얼 호스트를 설정한다. 설정되어 있으면 자동으로 활성화된다.
<VirtualHost *80> </VirtualHost>	버츄얼 호스트 정보를 입력한다.

3 로그 이해 및 분석

1 access_log 확인

1. 아파치 설치 디렉터리 확인(분산 로그 vs. 중앙 로그)
   - vi /etc/httpd/logs/access_log



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

2 예) 첫 행 내용 확인

127.0.0.1 - - [12/May/2013: 11: 32: 18 +0900] "GET / HTTP/1.1" 403 4609 "-"
① ② ③ ④ ⑤ ⑥ ⑦
"Mozilla/5.0 (X11; Linux i686; rv: 12.0) Gecko/20100101 Firefox/ 12.0"
⑧

1) 서버에 요청을 한 클라이언트(원격 호스트)의 IP 주소
2) 클라이언트의 사용자 이름
3) 접속을 시도한 날짜와 시간
4) 요청한 메소드(GET)와 접근 경로(/), 프로토콜 종류(HTTP 1.1)
   : Get vs. Post, HTTP 1.0 vs. HTTP 1.1

1 access_log 확인

2 예) 첫 행 내용 확인

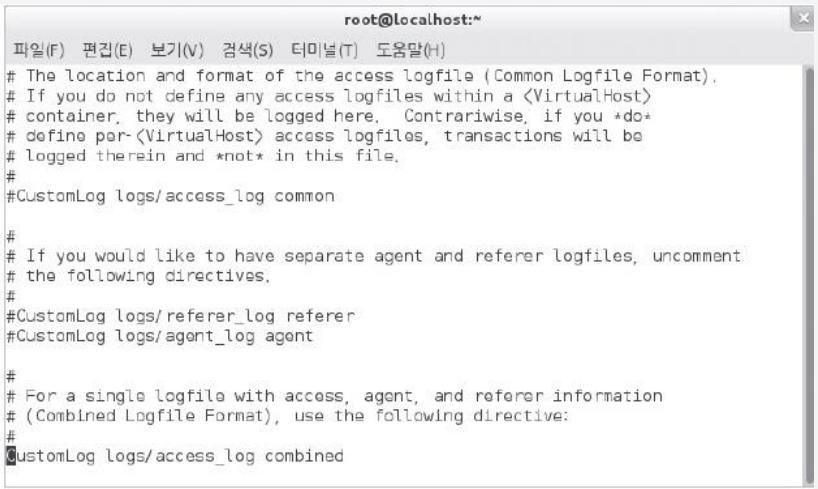
127.0.0.1 - - [12/May/2013: 11: 32: 18 +0900] "GET / HTTP/1.1" 403 4609 "-"
① ② ③ ④ ⑤ ⑥ ⑦
"Mozilla/5.0 (X11; Linux i686; rv: 12.0) Gecko/20100101 Firefox/ 12.0"
⑧

5) 서버가 클라이언트에 보내는 상태 코드
6) 클라이언트에 보내는 내용의 크기
7) 링크를 설정한 항목에 대한 접근 기록
8) 서버에 접근하는 클라이언트의 웹 브라우저 종류

1 access_log 확인

3 CustomLog

- 접근 로그(access_log)의 위치와 내용을 지정

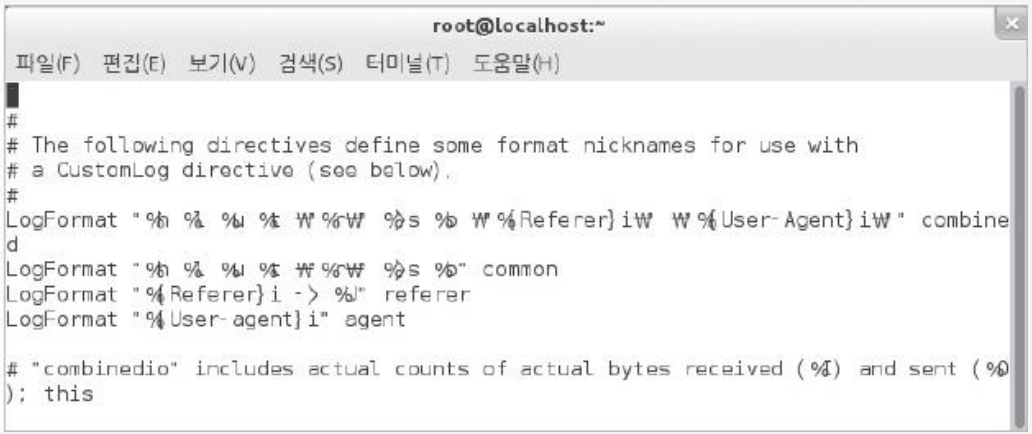


※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

3 LogFormat

- 일반적으로 사용하는
  로그 포맷은 combined와 common



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

3 LogFormat

인수	설명
%a	클라이언트의 IP 주소이다.
%A	로컬 IP 주소이다.
%b	헤더 정보를 제외하고 전송된 데이터의 크기로, 전송된 데이터의 크기가 0일 때 '-'로 표시한다.
%c	응답을 마쳤을 때의 연결 상태이다. X: 응답을 마치기 전에 연결이 끊김 +: 응답을 보낸 뒤에도 연결이 지속됨 -: 응답을 보낸 뒤 연결이 끊김
%{Header}e	환경 변수 헤더의 내용이다.
%f	요청된 파일 이름이다.
%h	클라이언트의 도메인 또는 IP 주소이다.
%H	요청 프로토콜의 종류이다.
%l	클라이언트 쪽에서 identd를 실행하고 있을 때 클라이언트의 로그인 이름이지만 %u와 마찬가지로 100% 신뢰할 수는 없다.

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

3 LogFormat

%m	요청 메소드이다.
%p	서버가 요청을 받아들이는 포트 번호이다.
%P	요청을 처리하는 자식 프로세스의 아이디이다.
%q	쿼리에 사용된 문자이다.
%r	요청의 첫 번째 줄이다.
%s	서버가 클라이언트에 보내는 상태 코드이다. 이 정보는 요청이 성공했는지(2로 시작하는 코드), 클라이언트에 오류가 있는지(4로 시작하는 코드), 서버에 오류가 있는지(5로 시작하는 코드)를 알려주므로 매우 중요하다.
% {format}} t	웹 서버에 작업을 요구한 시간이다.
%T	웹 서버가 요청을 처리하는 데 걸린 시간(초)이다.
%u	클라이언트의 사용자(상태 코드 401을 돌려줄 경우 등록되지 않은 사용자)로 사용되지만 100% 신뢰할 수 있는 데이터는 아니다.
%U	요청된 URL 경로이다.
%v	요청을 처리하는 서버의 이름이다.

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

4 SetEnvIf : Remote_Addr, Request_URI

- Customlog에서 env 인수를 이용하여 이름이 dontlog인 항목을 로깅하지 않도록 설정

```
SetEnvIf Remote_Addr "127₩.0₩.0₩.1" dontlog
CustomLog logs/access_log common env =! dontlog
```

- Request_URI가 ₩.ida로 요청하는 것을 'worm' 이라고 이름 짓고, 그 로그를 env를 이용 하여 로깅하지 않게 설정

```
SetEnvIf Request_URI ₩.ida worm
CustomLog logs/access_log common env =! worm
```

1 access_log 확인

4 SetEnvIf

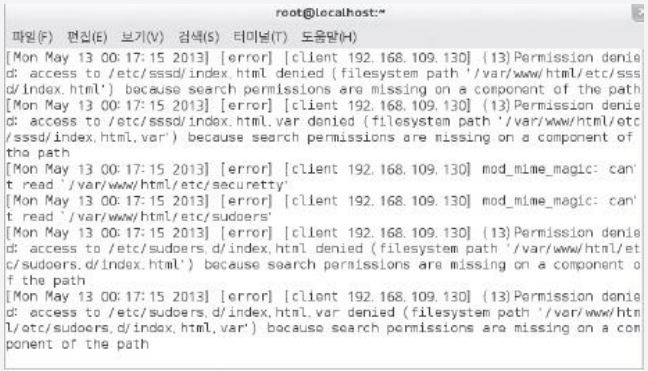
구분		내용
요청 헤더	Host	호스트 이름이다.
	User-Agent	클라이언트의 웹 브라우저 버전이다.
	Referer	링크된 경로이다.
	Accept-Language	접속자가 사용하는 언어(en, kr 등)이다.
요청 특징	Remote_Host	요청하는 클라이언트의 호스트 이름이다.
	Remote_Addr	요청하는 클라이언트의 IP 주소이다.
	Server_Addr	요청을 받는 서버의 IP 주소(Ver 2.0.43 이후)이다.
	Request_Method	사용한 메소드 이름(GET, POST 등)이다.
	Request_Protocol	요청한 프로토콜 이름과 버전(HTTP 1.1 등)이다.
	Request_URI	HTTP에서 요청한 자원이다.

※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

5 에러 로그(error_log)

- /etc/httpd/logs/error_log에서 에러 로그 내용 확인
- httpd.conf 파일의 ErrorLog 값을 이용해 경로 설정



※ 출처 : 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017

1 access_log 확인

6 에러 로그 마지막 줄

Mon May 13 00: 17: 15 2013] [error] [client 192.168.109.130] (13)Permission denied:
① ② ③ ④
access to /etc/sudoers.d/index.html.var denied (filesystem path'/var/www/html/
④ ⑤ ⑥
etc/sudoers.d/index.html.var') because search permissions are missing on a component
⑥
of the path
⑥

1 access_log 확인

6 에러 로그 마지막 줄

1) 오류가 생긴 날짜와 시간
2) 오류의 심각성을 알리는 것(debug, info, notice, warn, error, criti 등으로 표시) (error level)
3) 클라이언트의 접근 IP 주소
4) 접근한 웹 경로
5) 성공 또는 실패 여부
6) 발생한 오류에 대한 설명

4 전자화폐

1 전자화폐 시스템

- 전자화폐
  - 네트워크를 통해 전자적으로 구현된 화폐가치를 교환하여 결재를 수행하는 시스템
  - 결재 방법과 사용 방법에 따라 IC카드형과 네트워크형으로 분류

1 전자화폐 시스템

- 전자화폐

IC카드형 전자화폐

- IC칩을 포함하고 있는 플라스틱 카드에 화폐가치를 저장한 후 결재 시에 이를 인출하여 사용하는 방식
- 외형상 신용카드와 동일

1 전자화폐 시스템

전자화폐

IC카드형 전자화폐

- 사용자는 네트워크나 은행의 ATM 기기를 통하여 자신의 계좌로부터 일정 금액을 인출한 후 이를 IC칩에 저장하여 사용

장점	단점
휴대가 간편하고 다양한 용도로 사용이 가능	시스템 구축 비용이 많이 들고 호환성에 문제

1 전자화폐 시스템

전자화폐

네트워크형 전자화폐

- 은행의 계좌로부터 인출된 현금에 상응하는 화폐가치를 네트워크를 통해 다운로드 하여 사용자의 컴퓨터에 저장하거나 인터넷상의 가상은행에 저장한 후 결재 시에 이를 인출하여 사용하는 방식

장점	단점
초기 구축 비용이 저렴	휴대가 어려움

1 전자화폐 시스템

- 전자화폐
  - 액면 가치를 보증
    - 은행이 서명한 디지털 신호로 표현된 가치 정보
  - 전자지불 시스템의 지불 서버와 같은 지불 브로커 없이 독립적인 구조로 결재를 수행하는 신용 기반
  - 사이버 공간에서 현금 개념으로 통용되는 전자적 지불 수단

1 전자화폐 시스템

- 고객, 상점 서버, 금융기관 등으로 구성(지불 서버 없음)
- 전자화폐의 장점
  - 이용자의 프라이버시를 지킬 수 있음
  - 지불을 위하여 요구되는 부가적인 수수료 등의 비용을 줄일 수 있음
  - 개방성과 유통성을 실현 가능

1 전자화폐 시스템

- 전자화폐의 안전성
  - 사전조치 메커니즘
    - 암호기술, 인증기술, 제한 한도액 규제, 인증제도 등
  - 중간 대응책
    - 추적가능성과 모니터링, 중앙시스템과의 조회, 거래이력의 보존과 온라인 검증, 정보개시 등
  - 사후조치
    - 장치의 사용거부, 시스템의 정지 등

1 전자화폐 시스템

- 전자화폐의 정보보호 요구조건

디지털 정보의 완전 독립성(independence)

  - 화폐의 정당성을 인증 받기 위한 은행의 서명,
    복사방지를 위한 기술 등

1 전자화폐 시스템

전자화폐의 정보보호 요구조건

전자화폐의 재사용 불가능

- 복사 및 위조 등으로 인한 부정사용을 할 수 없도록 하기 위함
- 보완 대책
  - 위, 변조 방지를 위한 마이크로칩을 이용한 안전장치를 내장
  - 고성능 암호처리 프로토콜 설치
  - 전자화폐 발행 은행의 지속적인 모니터링
  - 전자화폐 거래 관련 기록 유지

1 전자화폐 시스템

전자화폐의 정보보호 요구조건

전자화폐의 익명성

- 정당한 사용자의 화폐 사용 내역은 알려져서는 안 됨
- 사용자의 사생활은 보호되어야 할 뿐만 아니라 사용자의 구매내역 등이 추적 불가능해야 함

거래의 오프라인 처리(실제 거래)

- 사용자와 상점 사이에서의 거래는 오프라인 방식으로 처리가 이루어 져야 함

1 전자화폐 시스템

- 전자화폐의 정보보호 요구조건

타인에게 양도 가능

  - 전자화폐를 받은 상점이나 사용자는 다시 해당 전자화폐를 다른 상점이나 제 3의 사용자에게 사용이 가능해야 함

1 전자화폐 시스템

전자화폐의 정보보호 요구조건

부정적인 사용자의 경우 익명성 취소

- 부정적인 사용자의 경우 익명성 취소가 요구
- 부정적 사용
  - 돈세탁, 돈 약탈, 마약구매, 무기구매 등
- 익명성 취소
  - 전자화폐의 소유자를 식별하는 소유자 추적
  - 은행으로부터의 화폐인출을 식별하기 위한 화폐 추적

2 전자화폐 지불 시스템

- IC 카드 기반 전자화폐 지불 시스템

VISA Cash Card

- IC카드형 전자화폐
- 1996년 Visa International사가 개발
- 종류: 일회용, 재충전용
- 현재 미국, 일본, 호주 등지에서 시범적으로 사용
- 개인 간 자금이체는 불가능하고, 자금거래가 정산기구를 통해 이루어짐

2 전자화폐 지불 시스템

- IC 카드 기반 전자화폐 지불 시스템

Mondex Card

- 영국 몬덱스(Mondex)사가 개발
- IC카드형 전자 화폐인 몬덱스 카드(Mondex Card)는 신용카드 크기의 플라스틱 카드에 IC칩을 내장한 것

2 전자화폐 지불 시스템

- IC 카드 기반 전자화폐 지불 시스템

Mondex Card

- 특징
  - 표면에 화폐가치를 판독하기 위한 금속제의 접점이 부착
  - 전자화된 가치를 입력 또는 인출하여 다양한 결제에 이용

2 전자화폐 지불 시스템

- 네트워크 기반 전자화폐 지불 시스템

E-Cash

- 네트워크형 전자화폐
- 네덜란드의 DigiCash사가 1994년 10월에 개발한 인터넷상의 전자화폐
- 네트워크상에서 무상으로 제공되는 Digital Wallet이라는 소프트웨어를 설치함으로써 E-Cash에 의한 결제 가능
- 전자가치: 특별히 고안된 전자지폐로 표시되며 발행인의 전자서명으로 확인

2 전자화폐 지불 시스템

- 소액 지불 시스템

Millicent

- Digital Equipment Corporation이 1/10센트(0.001 달러) 정도의 소액지불도 가능하도록 설계한 분산(decentralized) 소액지불시스템
- 지불은 제 3자와의 접촉 없이도 상인의 사이트에서 효율적으로 확인 가능

2 전자화폐 지불 시스템

- 소액 지불 시스템

PayWord

- MIT Laboratory for Computer Science의 Ron Rivest와 이스라엘 Weizmann Institute of Science의 Adi Shamir가 개발한 크레딧-기반의 소액지불시스템
- 좀 더 빠른 해쉬 함수(hash function)를 이용하여 지불 당 소요되는 공개-키 동작의 수를 감소시키고자 하였음

2 전자화폐 지불 시스템

- 소액 지불 시스템

Micro Mint

- PayWord를 개발하였던 Ron Rivest와 Adi Shamir의 두 번째 소액지불시스템
- 공개-키 암호화를 필요로 하지 않는 독특한 형식의 전자화폐에 기반