제 3 장 어플리케이션 보안

1. 인터넷 응용 보안

1.1 FTP 보안

1.1.1 FTP 개념

o 핵심가이드

- FTP 프로토콜의 이해
- FTP에서 사용되는 두 개 포트의 접속 방식 이해
- FTP 주요 명령어 및 서버 응답 유형의 이해

FTP (File Transfer Protol)는 인터넷상의 컴퓨터들 간에 파일을 교환하기 위한 표준 프로토콜이다 (IETF RFC 959). 화면에 표시할 수 있는 웹 페이지와 관련 파일들을 전송하는 HTTP (Hypertext Transfer Protocol), 전자우편을 전송하는 SMTP (Simple Mail Transfer Protocol)등과 같이, FTP도 역시 인터넷의 TCP/IP 응용 프로토콜 중의 하나이다.

사용자 입장에서는 원하는 파일의 전송을 위해 간단한 명령어를 통하여 FTP를 사용하거나, 또는 그래픽 사용자 인터페이스를 제공하는 상용 프로그램을 쓸 수도 있다. 웹 브라우저에서도 선택한 프로그램을 다운로드 할 때 FTP를 사용할 수 있다. 또한 FTP를 사용하여 서버에 있는 파일을 지우거나 이름을 바꾸거나 옮기거나 복사하는 등 갱신 작업을 할 수도 있다.

FTP 보안을 구현 하여 안전한 FTP를 운용하기 위해서는 먼저 FTP 프로토콜의 개념 및 동작 원리를 이해하고, 기본적인 FTP 서비스 운영 실무 기술을 습득하여야 하며, FTP 서비스 운영에 있어서 주의 해야 할 공격 유형에 대해서 이해하고 각 공격 방법에 대한 대응 방법을 숙지해야 한다.

(1) FTP 프로토콜 이해

FTP 프로토콜의 개념을 학습함에 있어서 가장 중요한 것은 제어 연결과 데이터 연결의 차이점을 이해하는 것이다. FTP 프로토클의 큰 특징은 정상적인 서비스를 수행하기 위해 두 개의 포트를 사용한다는 것이다. 이로 인해 FTP 프로토콜은

active 모드와 passive 모드의 두 가지 접속 방식이 존재하게 되는데, 이들의 차이를 반드시 구별 할 줄 알아야 하겠다. 또한 FTP 명령어들과 이 명령들이 서버에 전송되었을 때 이 명령에 대한 응답도 서버에서 다양하게 전달하게 되는데 이들에 대해서도 이해 할 수 있어야 한다.

(가) 두 개의 Connection

파일을 전송하기 위해 두 개의 TCP 연결을 동시에 사용 한다.

1) 제어 연결 (control connection)
- o 클라이언트에서의 서버로의 명령과 서버의 응답을 위한 연결
- o 21 번 포트 사용
- o 21번 포트 : 명령 또는 응답형태의 제어정보를 전송,
- o 전체 FTP세션 동안 계속 연결 상태를 유지

2) 데이터 연결 (data connection)
- o 파일이 전송될 때 생성되는 데이터 연결
- o 20 번 혹은 1024 이후 포트 사용
- o 각각의 파일 전송 때마다 설정 되며 전송이 완료되면 폐쇄

(나) FTP 명령어(commands)
- o 3 또는 4 바이트의 ASCII 대문자로 구성
- o 옵션 변수를 갖는 경우 존재

1) 주요 FTP 명령
- o ABOR : 현재 전송중인 파일 전송 중단
- o CWD : 작업 디렉토리 변경
- o DELE : 원격지 파일 삭제
- o LIST : 원격지 파일 목록 보기
- o MDTM : 파일의 수정 시간 보기
- o MKD : 원격 디렉토리 생성
- o MODE : 전송 모드 변경
- o NLST : 원격 디렉토리 목록 보기

o NOOP : 아무 작업 안함
o PASS : 패스워드 전송
o PASV : passive 모드로 전환
o PORT : data 포트 열기
o PWD : 작업 디렉토리 표시
o QUIT : 연결 종료
o RETR : 원격지 파일 가져오기
o RMD : 원격지 디렉토리 제거
o SIZE : 파일 사이즈 리턴
o STOR : 원격지에 파일 저장
o USER : 사용자명 전송

(다) FTP 응답 (replies)

o 3 바이트의 ASCII 숫자와 숫자 뒤 옵션 메시지로 구성
[예] 331 Username OK, password required
125 Data connection already open: transfer starting
425 Can't open data connection

1) 주요 FTP 응답

o 200 명령 OK
o 500 구문 오류, 명령이 인식되지 않았음.
o 501 매개변수나 인자에서는 구문 오류.
o 502 명령이 구현되지 않았음.
o 110 재개시 표시기 응답.
o 119 단말기 사용 불가, 우편 박스 기능 시도.
o 211 시스템 상태, 또는 시스템 도움말 응답.
o 212 디렉토리 상태.
o 213 파일 상태.
o 214 도움말 메시지.
o 120 서비스가 nnn분 후에 준비됨.
o 125 데이타 연결이 이미 개방되어 있음; 전송 시작.

o 225 데이타 연결 개방; 진행중인 전송이 없음.
o 425 데이타 연결을 개방할 수 없음.
o 226 데이타 연결 폐쇄; 요청된 파일 동작이 성공적임.
o 227 수동 모드로 전환. h1, h2, h3, h4, p1, p2.
o 230 사용자가 로그인 되었음, 계속하시오.
o 530 로그인 되지 않았음.
o 331 사용자 명 ok, 비밀 번호가 필요함.
o 332 로그인을 위한 계정이 필요함.
o 532 파일 저장을 위한 계정이 필요함.
o 150 파일 상태 ok; 데이타 연결을 곧 개방할 것임.
o 152 알려지지 않은 사용자; 우편이 관리자에 의하여 회송될 것임.
o 250 요청된 파일 행위 ok, 완료되었음.
o 350 요청된 파일 행위가 더 이상의 정보를 보류하고 있음.
o 450 요청된 파일 행위가 이루어지지 않았음: 파일이 이용될 수 없음
o 550 요청된 행위가 이루어지지 않았음: 파일 사용 불가
o 451 요청된 행위가 강제 종료되었음: 국부 오류가 처리중임.
o 551 요청된 행위가 강제 종료되었음: 알려지지 않은 페이지 형식.
o 452 요청된 행위가 이루어지지 않았음: 시스템의 기억장치 공간이 불충분함.
o 552 요청된 파일 행위가 강제 종료 되었음: 기억장치 할당이 초과 되었음.
o 553 요청된 행위가 이루어지지 않았음: 파일명이 없음.

(라) Active 모드와 Passive 모드

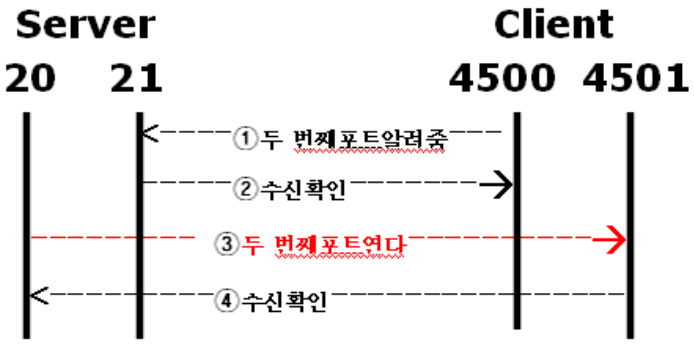
FTP 프로토콜은 두 개의 포트를 사용한다. 결과적으로 FTP는 서버와 클라이언트 사이에 두 번의 connection이 성립이 되어야 정상적인 서비스가 이루어 질 수 있다는 얘기인데, 이 두 번의 connection이 이루어지는 방식에 따라 active 모드와 passive모드로 나눌 수 있다.

이 두 가지 연결 방식의 장, 단점 및 작동 원리를 정확히 이해할 수 있어야 FTP 서비스의 접근제어를 위한 방화벽 설정을 효과적으로 적용 할 수 있으므로 반드시 숙지 하도록 한다.

한 가지 반드시 기억해야 할 점은 Active모드 및 Passive모드의 사용 여부는 FTP 서버가 아닌 클라이언트가 결정한다는 사실이다.

1) Active 모드

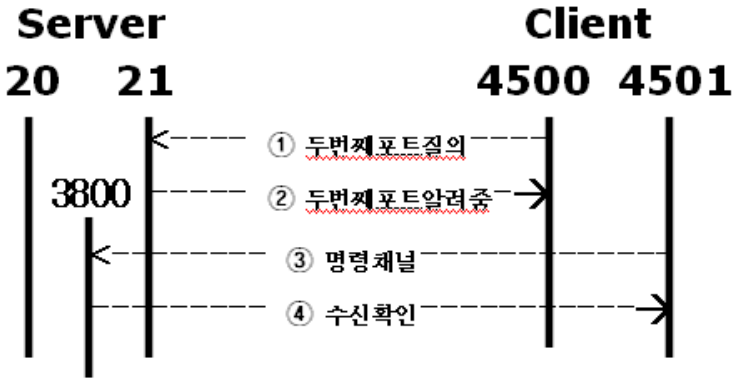
o 21, 20번 포트 사용
o 서버에서는 두 개의 포트만 열면 서비스 가능
o 두 번째 connection은 서버에서 클라이언트로 접속
o 클라이언트에 방화벽 설치시 접속 불가



(그림 3-1) Active 모드

2) Passive 모드

o 21, 1024 이후 포트 사용
o 데이터 전송 위해 1024번 이후 포트 사용
o 서버에서 클라이언트로 접속해야 하는 모순 해결 위해 고안된 방식
o 서버에서는 21번 포트와 1024번 이후의 모든 포트 오픈.
o 보안 위해 서버에서 passive모드로 사용할 포트 제한



(그림 3-2) Passive 모드

1.1.2 FTP 서비스 운영

o 핵심가이드

- 유닉스/리눅스 환경에서의 FTP 데몬(wuftp, proftp, vsftp) 운영 방법 숙지
- 윈도우즈 인터넷 정보 서비스 운영 방법 숙지
- 각 데몬들의 특징 이해
- 데몬들의 주설정 파일 이해 및 주요 보안 옵션 숙지

FTP 프로토콜에 대해 이해가 되었다면 실제 FTP 서비스를 운용함으로써 실무 기술을 습득 할 수 있다. 일반적으로 FTP 사이트(서버)는 FTP 서버 소프트웨어(데몬)를 사용하는 컴퓨터이다. UNIX 계열의 proftp, wuftp, 레드햇 리눅스의 vsftpd, 윈도우즈 NT 계열의 IIS 등은 잘 알려지고 많이 사용하는 FTP 서버 소프트웨어이다. 이들 데몬의 설치 방법 및 기본적인 운용 방법, 그리고 여러 가지 옵션들에 대해서 이해를 충분히 해두어야 하겠다.

(1) FTP 서버 설치 및 운영

(가) UNIX 환경에서의 FTP 서버 설치 및 운영

1) wuftp 설치 및 운영

가) 설치 패키지

o wu-ftpd : 기본 FTP 데몬
o anonyftp : 익명 접속 가능케 해주는 확장 패키지

나) 운영 사항

o 홈 디렉토리 상위로 이동하는 것 차단
o anonymous 계정 사용 하기
o 접근 방법 설정
o connection timeout 지정
o 로그 활용

2) proftpd 설치 및 운영

가) proftpd 특징

- o http://www.proftpd.org/
- o wu-ftpd의 대안으로 개발
- o 매우 안정하고 빠름
- o xinetd / standalone 형태로 작동 가능

나) ftp접속시 확인 설정

- o /etc/passwd, /etc/shadow에 사용자 계정이 있는지 검사.
- o /etc/ftpusers에 사용자 id가 있으면 거부.
- o /etc/shell에 등록되지 않은 쉘을 사용하는 유저는 접근 거부

RequireValidShell off

다) proftpd 설정 파일 옵션

- o ServerType standalone (inetd) : 서버 타입 설정
- o RootLogin off : 루트 계정 로그인 허용 안함
- o User nobody : 데몬 동작 계정
- o Group nobody : 데몬 동작 그룹
- o ServerIdent On "Welcome to FTP" : 버전 정보 숨기기
- o MaxClients : 최대 접속 허용
- o TimeoutLogin : 아이디/암호로 인증이 완료 될 때까지의 제한 시간
- o TimeoutIdleftp : 접속 후 아무런 데이터 전송이 없는 idle 상태
- o TimeoutSession : 일정 시간 후에는 무조건 접속 종료
- o Limit Command : 사용 가능한 command를 제한

3) vsftpd 설치 및 운용

가) 주요 기능

- o 가상 IP별 별도의 환경 설정 기능 (설정파일의 listen_address= 이용)
- o 가상 사용자 설정
- o 전송 대역폭 지정
- o PAM 지원 (버전 1.2.0부터는 PAM을 통한 wtmp에 로긴 로그를 남김)

o xferlog 표준 로그 파일보다 상세한 자체 로그 파일 형식 지원
o Standalone 방식과 inetd(xinetd)를 통한 운영 모두 지원
o IP별 다른 환경 파일 지정 기능 (tcp_wrappers와 함께 사용할 때)

나) 설치
o http://vsftpd.beasts.org/

다) 운용 사항 및 서버 옵션
o anonymous 계정 사용
o 서버 동작 모드 결정
o 전송 속도 제한
o 최대 접속 설정
o 파일 읽기, 쓰기 모드 결정
o 서버 접속 로긴 메시지 표시
o 로그 파일 (파일 전송 로그 및, 포맷 결정)

(나) 윈도우즈 환경에서의 FTP 서버 설치 및 운영

1) IIS FTP 설치 및 운용

가) 서버 설치 및 운용
o IIS 하위 구성 요서 선택
o 공용 파일, 인터넷 정보 서비스 스냅인, File Transfer Protocol(FTP) 서버
o 관리도구 - 인터넷 서비스 관리자 선택
o 두 이상의 NIC 및 IP 환경에서 서버 운용하기
o 운용 포트 변경
o 연결 및 연결 시간 제한
o 보안 계정 사용 및 익명 연결 허용
o FTP 메시지 지정
o 홈 디렉토리 지정 및 디렉토리 보안 설정
o 가상 디렉토리 사용

1.1.3 FTP 공격 유형

o 핵심가이드

- FTP 공격 유형의 이해

FTP 서비스를 운영함에 있어 주의해야할 공격 유형은 bounce attack, tftp공격, anonymous ftp 공격, ftp 서버 자체 취약점 공격 등이 있으며, 특히 FTP 서버 자체 취약점에 대한 공격은 최신 해킹 경향에 따라 끊임 없이 공격 기법이 발견 되므로, 항상 최신 보안 경향에 관심을 기울이고, 주기적인 패치에 힘써야 한다.

(1) bounce attack

- o 익명 FTP 서버를 이용해 그 FTP 서버를 경우해서 호스트를 스캔
- o FTP PORT 명령을 이용
- o FTP 서버를 통해 임의의 네트워크 접속을 릴레이함으로써 수행
- o 네트워크를 포트 스캐닝하는데 사용

(2) tftp 공격

- o trivial FTP
- o FTP보다 간단하지고 기능이 조금 덜한 네트웍 애플리케이션
- o 자체 디스크를 가지고 있지 않은 머신들(X-터미널등)이 부팅 시에 필요한 자료와 정보를 받아오기 위해서 사용
- o 디렉토리를 보여주지 않아도 되는 경우에 사용
- o 사용자 인증이 불필요
- o 불필요한 정보 또한 유출될 수 있는 기회를 제공
- o TCP 대신에 UDP를 사용, 69번 포트
- o 표준 문서 rfp 1350

(3) anonymous ftp

- o 보안 절차를 거치지 않은 익명의 사용자에게 FTP 서버로의 접근 허용
- o 익명 사용자가 서버에 쓰기 권한이 있을 때 악성 코스 생성 가능

(4) ftp 서버 자체 취약점

- o wuftp 포맷스트링 취약점
- o 각종 버퍼 오버플로우 공격 가능

(5) 스니핑에 의한 계정 정보 및 메시지 유출 위험

o ID, password 입력 후 접속 시도 할 때 정보의 암호화가 이루어지지 않음

o 네트워크 스니핑 공격에 취약

o ftp 세션 자체는 암호화가 되어 있지 않아 프라이버시 보호 기능 제공 않음

(6) brute force 공격에 의한 계정 유출

o 무작위 대입법

1.1.4 FTP 보안 대책

o 핵심가이드

-각 FTP 공격 유형에 대한 대응 방법 숙지

(1) anonymous FTP 보안 대책

o anonymous 사용자의 루트 디렉토리, bin, etc, pub 디렉토리의 소유자와 퍼미션 관리

o $root/etc/passwd파일에서 anonymous ftp에 불필요한 항목 제거

(2) tftp 보안 대책

o tftp가 불필요한 경우 제거

o tftp가 필요한 경우 secure mode로 운영

(3) 서버 자체 취약점 대책

o 최신 ftp 서버 프로그램 사용 및 주기적인 패치

1.2 MAIL 보안

mail서비스는 인터넷 환경에서 필수불가결한 응용 서비스이면서 많은 보안상의 과제를 파생시키는 분야이다. mail보안에 있어서는 먼저 mail 서비스를 구성하는 핵심 프로토콜인 smtp프로토콜의 개념을 이해하고, 사용자 기반의 메일 서비스를 위한 pop, imap프로토콜의 개념을 이해해야 한다. 이러한 기본 개념 하에서 각 응용서비스를 실제 운용할 수 있는 mail서버와 pop서버의 설치 및 운영 실무지식이 필요하다. 이를 바탕으로 mail서비스 운영에 있어서 주의해야할 보안상 문제에 대해서 이해하고 각 문제들을 해결할 수 있는 대책을 숙지하여야 한다.

1.2.1 mail 개념

mail서비스를 구성하는 핵심 프로토콜에는 MTA간의 직접 메일 전송과 전달과정을 제어하는 smtp프로토콜과 MTA와 MUA간의 사용자 기반의 메일 서비스를 위한 pop, imap프로토콜이 있다. [1,2급 공통출제]

o 핵심가이드

- 메일 서비스를 위한 프로토콜에 대한 이해
- 메일 헤더에 대한 이해
- 메일 헤더 분석

(1) smtp 프로토콜

Simple Message Transfer Protocol로서 전자우편을 보내고 받는데 사용되는 기본 TCP/IP 프로토콜이다.

o MTA, MDA, MUA의 개념
  - MTA : 메일을 전송하는 서버
  - MDA : MTA에게서 받은 메일을 사용자에게 전달
  - MUA : 사용자들이 사용하는 클라이언트 어플리케이션

o 메일 헤더 구조
  - Received
  - From

- Reply-to
- To

o MX Records의 이해
- Mail Exchange
- 서버에서 오는 메일을 처리할 메일 서비스 지정
- MX record 중복 지정 및 MX 뒤의 숫자에 대한 이해

o smtp 명령어의 이해
- HELO, EHLO, MAIL, RCPT, DATA
- RSET, NOOP, QUIT, HELP, VRFY
- EXPN, VERB, ETRN, DSN, AUTH

(2) pop 프로토콜

Post Office Protocol로서 메일서버가 사용자를 위해 전자우편을 수신하고 그 내용을 보관하기 위해 사용되는 클라이언트/서버 프로토콜이다.

o 클라이언트에서 POP3 데몬을 이용하여 메일을 직접 내려 받아 읽어옴
o POP3 데몬 포트 : 110번 포트
o POP3를 이용해 메일 서버에서 가져온 메일은 더 이상 서버의 메일 박스에 남아 있지 않으므로 사용자가 고정적인 위치에서 메일을 받는 경우에 유리

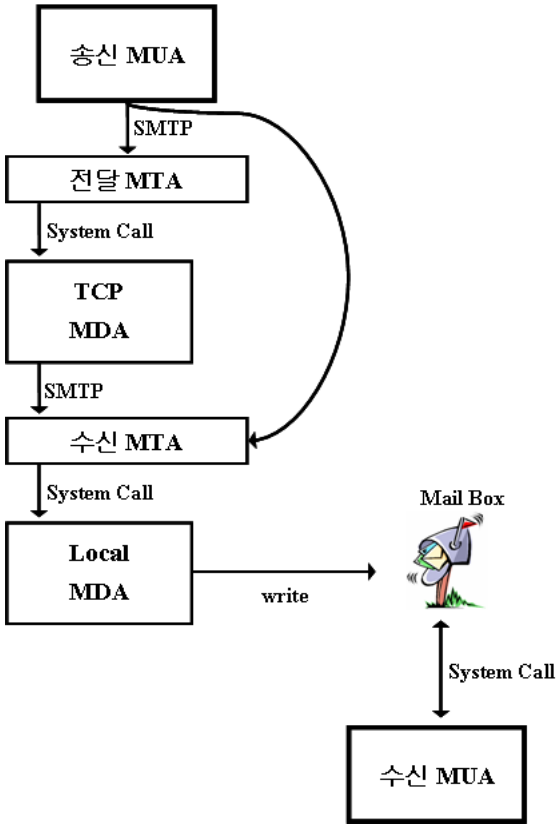
(3) imap 프로토콜

Internet Message Access Protocol로서 메일서버를 이용하여 전자우편을 수신하고 보관하는 클라이언트/서버형 프로토콜이다.

o IMAP의 경우 143번포트 사용, IMPA3의 경우 220번포트 사용
o pop3와 같은 역할을 하지만 메일을 보내는 방법에 차이가 있음
o IMAP로 접속하여 메일을 읽으면 메일 서버에는 메일이 계속 존재(메일 헤더만 보고 읽을 수 있으며, 읽은 메일은 읽지 않은 메일과 구분되어 표시)
o POP3는'보관하고 전달하는'서비스라고 비유할 수 있으며, IMAP은 원격지 파일 서버라고 비유할 수 있음

(4) 메일 헤더 분석 및 역추적 방법

o 메일 전송 과정



(그림 3-3) 메일 전송 과정

o 메일 헤더 분석

o IP주소 질의 및 추적 방법

1.2.2 mail 서비스 운영

mail 서비스를 실제로 운영함에 있어서는 각 플랫폼별 mail서버 어플리케이션(MTA 및 pop서버)의 설치와 운영지식을 다양한 실무 경험을 통해 익히는 것이 필요하다. [1,2급 공통출제]

o 핵심가이드

- 메일서버 설치 및 운영 기법
- pop서버 설치 및 운영 기법

(1) mail서버 설치 및 운영

o sendmail 설치 및 운영
o qmail 설치 및 운영
o MS exchange 설치 및 운영
o 메일 로그 설정
o 메일 용량 제한
  - 전체 용량 제한 설정 방법 (디스크 쿼타)
  - 한 번에 받을 수 있는 최대 용량 크기 제한 설정 방법 (MaxMessageSize)

(2) pop 서버 설치 및 운영

o qpopper 설치 및 운영
o qmail과 vpopmail 연동 설치 및 운영
  - 가상 도메인과 가상 유저에 대한 이해

1.2.3 mail 서비스 공격 유형

mail 서비스에 있어서 공격 유형은 크게 메일 사용자 클라이언트의 취약점을 이용한 사용자 컴퓨터 공격과 메일 서버(MTA)의 취약점을 이용한 메일서버 공격으로 나뉜다. mail 서비스 관련 공격은 최신 공격 경향에 따라 끊임없이 새로운 공격 기법이 발견되므로, 항상 최신 보안 경향에 관심을 갖고 있어야 한다. [1,2급 공통출제]

o 핵심가이드
- 메일 클라이언트에 대한 공격 유형의 이해
- 메일 서버에 대한 공격 유형의 이해

(1) 메일 클라이언트(outlook 등) 최신 취약점을 이용한 공격 기법

o Active Contents 공격
o 트로이 목마 공격

(2)메일 서버(sendmail 등) 취약점을 이용한 메일 서버 공격 기법

o 버퍼 오버플로우 공격
o 서비스 거부 공격

1.2.4 spam 대책[1급]

mail 서비스에 관련된 중요한 자원관리 문제로서 spam 메일에 대한 대책은 매우 중요하다. 이는 사용자 관점에서 클라이언트 어플리케이션을 이용한 spam 대책과 mail 서버 관리자 과점에서의 spam relay 차단 대책으로 나눌 수 있다.

o 핵심가이드
- 클라이언트 관점에서의 spam 메일 대책
- 메일 서버 관리자의 관점에서의 spam 메일 대책
- 스팸 블랙리스트 기법에 대한 이해
- 스팸 필터링 어플리케이션의 운영 방법

(1) 사용자 관점에서의 spam 메일 대책

o 메일 클라이언트 프로그램을 이용한 메일 필터링
o spam 대응 조치 요령

(2) mail 서버 관리자의 spam relay 대책

o 메일서버에서의 spam 릴레이 허용 불가 설정
  - sendmail에서의 anti-spam 기능과 Acess DB를 이용한 스팸 릴레이 차단
  - qmail에서의 스팸 릴레이 차단
  - MS exchange에서의 기능설정이나 레지스트리 설정을 통한 스팸 릴레이 차단
  - EMWAC 메일서버에서의 스팸 릴레이 차단
o SMTP AUTH를 통한 relay 차단
  - sendmail.cf에서의 보안 설정

(3) SPF(Sender Policy Framework)

허용된 도메인이나 IP등에서 발송된 것인지 확인하는 방법

(4) RBL(Real-time Spam Black Lists)

- o SBL(Spamhaus block list)
- o DNSRBL(Domain Name System Real-time Black List)
- o URIBL(Real-time URI Blacklist)
- o SURBL(Spam URI Real-time Blocklists)

(5) procmail

- o MDA로서의 procmail
- o sendmail과 procmail을 이용한 메일 필터링
- o procmailrc
- o procmailrc 환경 설정
  - 환경변수
  - 처방(Recipes)
  - 플래그(Flag)

(6) Sanitizer

- o sanitizer의 특징
  - procmail ruleset
- o sanitizer 옵션의 이해
  - 확장자를 이용한 필터링 기능
  - MS Office 매크로에 대한 검사기능
  - 악성 매크로에 대한 score 기능
  - 감염된 메시지 보관 장소 설정 기능
  - <STYLE> 태크 변경 기능

(7) inflex

o Contents Scanner
o 첨부파일만 필터링 가능
  - MS-DOS Executables
  - PC Bitmap Data
  - AVI movies
  - MPEG movies
  - WAVE type audio file
o 새로운 파일 타입 추가 설정 기능
o 파일 이름을 기준으로 필터링

(8) SpamAssassin

o SpamAssassin 개념
  - 텍스트를 이용한 분석
  - 실시간 블랙리스트(Real-time Black List)
o SpamAssassin 의 스팸 분류 기준
  - 헤더 검사
  - 본문 내용 검사
  - 베이시언 필터링
  - 주요 스팸 근원지/비근원지 자동 생성
  - 주요 스팸 근원지/비근원지 수동 생성
o SpamAssassin 설치 및 설정 방법
  - SpamAssassin과 sendmail 연동
  - SpamAssassin과 qmail 연동
o SpamAssassin 환경 설정 방법
  - 최소 기준 스코어
  - 언어 제한
  - 메일 헤더 변경
  - 실시간 블랙리스트와의 연동

1.2.5 악성 mail 및 웜 대책[1급]

최근 급증하고 있는 e-mail을 이용한 바이러스 등의 악성코드가 포함된 악성 메

일에 의한 피해가 급증하고 있으며 이를 차단하기 위한 대책은 매우 시급하고 중요한 보안 문제이다. 특히 메일 클라이언트 응용프로그램을 목표로 하는 최신 공격 경향에 따라 끊임없이 새로운 악성 메일이 발견되므로 항상 최신 보안 경향에 관심을 갖고 있어야 한다.

o 핵심가이드

- 라우터를 이용한 대응 방법
- MTA에 의한 차단 방법
- Viruswall을 이용한 차단 방법

(1) 악성 메일 및 웜 대책

o 라우터에서의 악성 메일 및 웜 차단
  - class-map을 이용한 차단
  - policy-map을 이용한 차단
o 메일 서버 프로그램(MTA)에서의 패턴 매칭에 의한 차단
o Virus wall을 이용한 차단

1.2.6 mail 보안 기술[1급]

전자우편은 모든 분산 환경에서 가장 많이 사용하는 네트워크 기반 응용이며 모든 시스템 구조와 제품에서 광범위하게 사용되는 분산 응용이기도 하다. 전자우편의 폭발적인 증가 추세를 볼 때 인증과 기밀성 서비스에 대한 요구가 증가하고 있으며 현재 PGP와 S/MIME이 많이 사용되고 있다.

o 핵심가이드

- PGP 프로그램의 동작방식과 주요기능 이해
- PGP 프로그램을 이용한 메일 보안 기법 이해
- S/MIME의 기능 및 특징

(1) PGP(Pretty Good Privacy)

인터넷 전자우편을 암호화하고 복호화 하는데 사용되며 송신자의 신원을 확인함

으로써 메시지가 전달 도중 변경되지 않았음을 확인할 수 있는 전자서명을 보내는 데에도 사용된다.

o PGP의 구성요소
  - 인증 받은 메시지와 파일에 대한 전자서명 생성과 확인 작업 지원
  - RSA와 Diffie-Hellman 등 공개키 생성 지원
  - 공개키 분배 및 취득
o PGP의 주요기능
  - 전자서명
  - 기밀성
  - 압축
  - 단편화와 재조립
o PGP 암호화와 키 연결 관계
  - 세션키 생성
  - 키 식별자(Key Identifier)
  - 키 링(Key Ring)
o 공개키 관리
  - 공개키 관리의 접근법
  - 공개키 철회
o PGP 프로그램 사용법
  - 개인키/공개키 생성 및 공개키 분배 방법
  - 본문 및 첨부파일 암호화/복호화 방법

(2) S/MIME(Secure/Multipurpose Internet Mail Extention)

RSA 암호화 시스템을 사용하여 전자우편을 안전하게 보내는 방법

o MIME
  - MIME 내용 타입(text/html 등)
  - MIME 전송 부호화
o S/MIME의 구성요소
  - RSA, DSA, Diffie-Hellman 공개키 암호화
  - 3DES, RC4, IDEA, DES, RC2 대칭 암호화

- X.509 버전3 인증서 지원

o S/MIME의 주요기능

- 봉인된 데이터(Enveloped data)
- 서명 데이터(Signed data)
- 순수한 서명(Clear-signed data)
- 서명과 봉인된 데이터(Signed and enveloped data)

1.3 Web 보안

인터넷 서비스 환경이 World Wide Web 환경으로 통합되어 가고 있다. Web 서비스는 다양한 아키텍쳐에 의하여 구동되며, 이에 따른 많은 보안상의 과제를 파생기키는 분야이다. Web 보안에 있어서는 먼저 Web 서비스를 구성하는 핵심 프로토콜인 HTTP 프로토콜의 개념과 구조 그리고 동작 방식에 대해 이해하고 강화된 웹 프로토콜인 SSL과 TLS에 대해서도 개념을 잘 이해해야 한다. 이러한 기본 개념하에서 웹서버의 설치 및 운영 실무 지식을 통해 응용서비스를 실제 운용할 수 있어야 한다.

이러한 지식을 바탕으로 최근 급속도로 증가하고 있는 웹서비스에 대한 공격 유형을 이해하고 이에 대한 대응방법을 숙지하여야 한다. 또한 XML 환경으로 발전해 가는 추세에 맞추어서 XML 기반의 웹 보안 기술에 대한 학습이 필요하다.

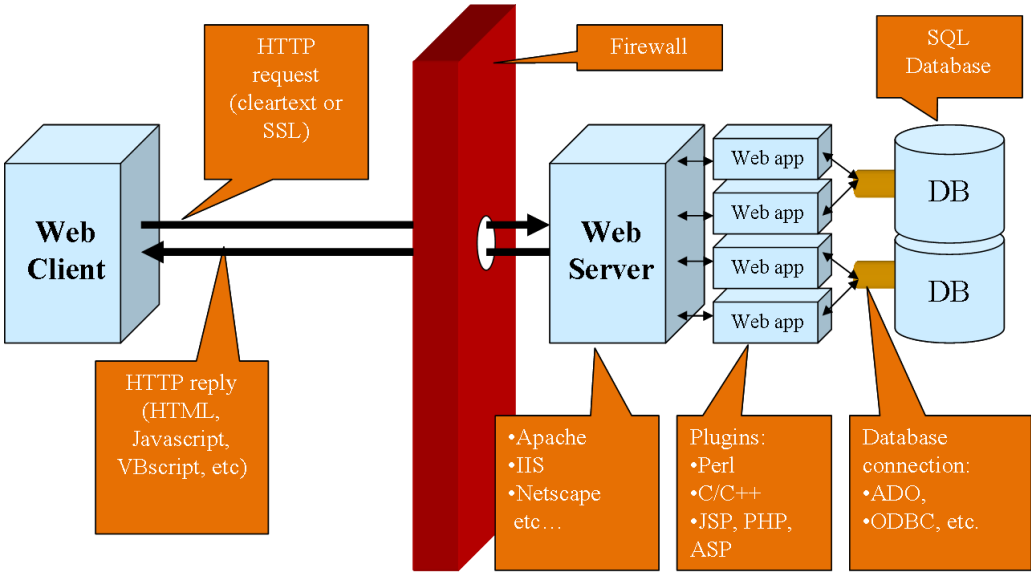
1.3.1 Web 개념

웹 어플리케이션의 아키텍쳐는 주요 전송 매개체인 HTTP(Hyper Text Transfer Protocol)을 사용하여 웹 서버와 웹 클라이언트 사이의 서비스 요청과 응답을 처리하게 된다. 웹 서버에서는 다양한 구조의 웹 어플리케이션 서비스를 제공하게 되며 가장 대표적인 웹 서버와 데이터베이스를 연동하여 정보서비스를 제공하는 것이다. 보안 측면에서 보다 향상된 웹서비스를 제공하기 위하여 SSL과 TLS 프로토콜 등이 사용된다. [1,2급 공통출제]

o 핵심가이드

- 웹 어플리케이션 아키텍쳐 이해
- 클라이언트와 서버간의 데이터 전송과정에 대한 이해

가. 웹 어플리케이션 아키텍처



(그림 3-4) 웹 서버와 웹 어플리케이션 동작 방식

나. 웹 프로토콜

o 핵심가이드

- HTTP 프로토콜 동작방식의 이해
- HTTP 프로토콜 요청방식의 특징

(1) HTTP/HTTPS

(가) HTTP 프로토콜의 동작방식

o HTTP Request
  - 1.0과 1.1의 차이점 이해
    · HTTP/1.1에서는 “?“를 이용하여 쿼리문을 만들 수 있다.
o HTTP Response
  - 응답의 나머지 부분의 형태를 결정하는 요소인 응답코드
  - 응답에 대한 추가적인 정보를 갖는 헤더 필드
  - 내용이나 응답의 몸체인 자료
o HTTP Headers의 의미 이해

- Hosts : 요청을 받는 서버의 호스트명이나 IP주소
- Accept : 허용 가능한 타입 (예, text/html, image/gif 등)
- Referer : 참조 URL
- User-Agent : 클라이언트의 웹 브라우저
- Cookie : 쿠키 정보
- Content-Length : POST 방식을 사용할 경우 동봉되는 데이터의 크기
- Content-Type : 동봉되는 데이터의 타입
- Cache-Control : 캐시값을 지정하지 않았을 경우특정 웹 리소스의 모든 캐싱을 방지

(나) HTTP 요청방식

o GET
- 요청 받은 정보를 검색

o HEAD
- GET 방식과 비슷하지만 요청 받은 자료를 되돌려 주지 않는다.

o POST
- 서버가 동봉된 정보를 받아들이고 서버에서 동작하도록 하는 요청

o OPTIONS
- 요청 받은 리소스에서 가능한 통신 옵션에 대한 정보 요청

o PUT
- 내용이 제공되는 리소스에 저장되기를 원하는 요청

o DELETE
- 명시된 리소스를 서버가 삭제될 것을 요청

o TRACE
- 루프백 메시지를 위한 요청을 송신

1.3.2 Web 서비스 운영

실제 웹 서비스를 운영하게 될 때에 대표적으로 사용하는 웹 서버 어플리케이션은 아파치와 IIS이다. 최근 웹 서비스에 대한 공격이 많아지고 있는 이유 중의 하나가 웹 서버의 기본 설정을 사용하는데 있으므로 기본적인 보안 설정은 필수불가결하다. [1,2급 공통출제]

o 핵심가이드

- 웹서버 설치 및 운영 기법
- 아파치 설정 보안
- 아파치 인증 및 접근제어
- 아파치 보안 모듈 사용
- IIS 웹서버 설치 및 운영
- IIS 설정 보안

(1) 아파치 웹서버 설치 및 운영

o 아파치 웹 서버 설치
  - --enable-so 옵션을 사용하여 각종 모듈을 사용할 수 있도록 설치
  - apxs를 이용한 모듈 생성 및 아파치 설정에 추가 방법
o 아파치 웹 서버 운영
  - 이름 기반의 가상호스트
  - IP 기반의 가상호스트

(2) 아파치 설정 보안

아파치 초기 설정 상태로 웹 서버를 운영할 경우 문제가 발생할 수 있거나 공격자에게 여러 정보를 주게 되므로 서비스하기 전에 보안과 관련된 설정을 적용해야 한다.

(가) 디렉토리 리스팅 제거

o 클라이언트가 인덱스 파일이 없는 디렉토리를 요청할 경우 디렉토리 내에 있는 파일들의 리스트가 열거되는 취약점
o Direectory 지시어 내에 Options의 Indexes 옵션 제거

(나) 서버의 정보 표시 제한 설정

o ServerTokens를 이용하여 정보를 제한

- ProductOnly : 웹 서버 종류만 표시
- Minimal : 웹 서버 종류와 버전 정보 표시
- OS : 웹 서버 종류와 버전, 운영체제 정보 표시
- Full : 웹 서버 종류와 버전, 운영체제, 설치된 모듈 정보 표시

o ServerSignature를 이용하여 정보를 표시 하지 않을 수 있음
- ServerSignature Off

(다) HTTP 요청방식 제한 설정

o Directory와 Limit 지시어를 이용한 요청방식 제한
- 파일의 업로드(PUT)와 수정(POST), 삭제(DELETE)를 패스워드 파일에 등록된 사용자만 이용 가능하도록 제한
- Require valid-user

(마) SSI 실행 제한 설정

SSI는 작은 양의 동적 컨텐츠를 생성시킬 때 사용하는 기능이다.

o SSI는 몇 가지 보안 위험 요소들이 존재한다.
- 부하 증가
- CGI 스크립트의 위험과 동일한 위험

o SSI 보안 확장
- SSI가 가능한 파일의 확장자 지정
  · AddType text/html .shtml
    AddHandler server-parsed .shtml
- SSI 페이지에서 스크립트나 프로그램의 실행 불가
  · Options 지시자에서 Includes를 IncludesNOEXEC로 교체

(바) CGI 스크립트 실행 제한 설정

CGI 스크립트를 어느 디렉토리에서나 실행할 수 있도록 할 경우 악의적인 사용자가 CGI 프로그램이 업로드하여 실행해서 임의의 명령을 실행시킬 수 있다.

o ScriptAlias 지시자에서 CGI 스크립트가 실행 가능한 디렉토리 지정

- ScriptAlias /cgi-bin/ "/var/www/cgi-bin"

o Options 지시자에 ExecCGI 옵션 명시 후 AddHandler로 CGI 스크립트로 처리할 파일의 확장자 지정

- Options ExecCGI
  AddHandler cgi-script .cgi

(사) 인증(Authentication)

o 기본 인증(Basic Authentication)

- 패스워드가 암호화 되어 저장되지만 클라이언트에서 서버로 전송되는 도중에는 암호화가 되지 않는다.
- 보호된 자원에 접속하는 매 순간마다 ID와 패스워드가 전송되지 때문에 쉽게 도청당할 수 있다.
- 사용법
  ・인증에 필요한 패스워드 생성
  htpasswd -c /usr/local/.passwd admin
  ・httpd.conf에 인증과 관련된 설정 추가

```
<Directory "/var/www/html/basic">
  AuthType Basic
  AuthName "Basic Authentication"
  AuthUserFile /usr/local/.passwd
  Require valid-user
</Directory>
```

o 다이제스트 인증(Digest Authentication)

- 네트워크 전송상에서 패스워드가 MD5 암호화 해쉬값으로 전송되기 때문에 기본 인증보다 안전하다.
- 아파치 설치시 --enable-auth-digest 옵션을 추가하여 설치해야 한다.
- 사용법
  ・인증에 필요한 패스워드 생성
  htdigest -c /usr/local/.digest kisa admin
  kisa는 영역을 의미하고 admin는 kisa의 영역에서 사용될 사용자의 이름
  ・httpd.conf에 인증과 관련된 설정 추가

```
<Directory "/var/www/html/digest">
  AuthType Digest
  AuthName "kisa"
  AuthDigestFile /usr/local/.digest
  Require valid-user
</Directory>
```

o 데이터베이스 인증(Database Authentication)

- 사용자 이름과 패스워드를 신속하게 확인할 수 있다.
- 서버에 다수의 사용자가 있을 경우 효율적이다.
- 아파치 설치시 --enable-auth-dbm 옵션을 추가하여 설치해야 한다.
- 사용법
  - 인증에 필요한 패스워드 생성

    dbmmanage /usr/local/.dbm adduser admin
  - httpd.conf에 인증과 관련된 설정 추가

```
<Directory "/var/www/html/database">
  AuthType Basic
  AuthName "DBM authetication"
  AuthDBMUserFile /usr/local/.dbm
  AuthDBMType GDBM
  Require valid-user
</Directory>
```

(아) 접근제어

o 영역/범위 지정 방식

- 디렉토리 지정 : <Directory> </Directory>
- 파일 지정 : <Files> </Files>
- URL 지정 : <Location> </Location>
- method 지정 : <Limit> </Limit>, <LimitExcept> </LimitExcept>
- 정규표현식 사용 가능한 방식
  - <DirectoryMatch>
  - <FilesMatch>
  - <LocationMatch>

o mod_access에 의한 접근제어

- order deny, allow
  · deny 지시자가 allow 지시자보다 먼저 검사된다. 접근은 기본적으로 허용
- order allow, deny
  · allow 지시자가 deny 지시자보다 먼저 검사된다. 접근은 기본적으로 차단
- order mutual-failure
  · allow 리스트에 있고 deny 리스트에 없는 호스트만 접근 허용
- satisfy
  · 사용자 인증과 접근제어를 동시에 사용할 경우 사용
  · any : 두 조건 중 하나만 맞으면 정상적으로 접근 가능
  · all : 두 조건이 모두 맞을 경우만 정상적으로 접근 가능

o 확장자별 접근제어

- 절대경로를 통하여 파일에 접근을 할 경우 특정 확장자로의 접근을 제한

o 환경변수별 접근제어

- 환경변수를 이용하여 특정 조건에서의 접근을 제한

(3) 아파치 보안 모듈 사용

o mod_rewrite

- 재작성 패턴을 받아들여 URL에 적용한다.
- Server Config, Virtual Host, Directory, .htaccess에서 사용 가능

o mod_setenvif

- 요청의 성격이 정규표현식에 해당하는지 여부로 환경변수를 설정

o mod_security

- 웹 서버 차원의 침입탐지 및 차단 기능
- 요청 필터링
- 회피 공격에 대응하기 위해 분석 전 경로나 파라미터 값 표준화
- POST 방식에 의해 전송되는 컨텐츠 검사
- SQL Injection, XSS, Buffer Overflow, Directory Traversal 대응

o mod_dosevasive

- DoS, DDoS 공격에 대한 대응
- 방화벽이나 라우터 등의 네트워크 장비와 통신 가능
- IP나 URI에 대한 동적인 해쉬 테이블 생성

(4) IIS 웹서버 설치 및 운영

o IIS 웹 서버 설치
o IIS 웹 서버 운영

(5) IIS 설정 보안

o HTMLA에 대한 접근 제어
o iisadmpwd 가상 디렉토리 제거
o 사용하지 않는 스크립트 파일 매핑 제거
o 디렉토리 리스팅 제거
o 익명 사용자 계정 권한 제한
o 인증 및 접근제어
  - 기본 인증
  - 다이제스트 인증
  - Windows 통합 인증
  - IP 주소나 도메인 이름에 대한 접근 제한 설정

1.3.3 Web 로그 보안

웹 서비스는 웹 서버와 웹 클라이언트 사이의 많은 개개의 서비스 요청과 서비스 응답으로 이루어지게 되므로, 웹 서버의 로그 관리를 철저히 하여 보안 문제 발생 시 이를 잘 분석하여 활용할 수 있어야 한다. [1,2급 공통출제]

o 핵심가이드

- 아파치 로그 관리 및 분석
- IIS 로그 관리 및 분석

(1) 아파치 로그 관리 및 분석

(가) 로그 파일 및 로그 관련 설정

o 로그 포맷

- 로그 포맷 중 불필요한 내용은 로그에 남지 않도록 설정

o 로그 분할

- 로그 파일의 크기가 커지면 웹 서버가 동작하지 않음
- rotatelog를 이용한 로그 분할
- TransferLog 지시어를 이용한 로그 분할

o ErrorDocument 지시어 이용

- 특정 상태코드를 반환할 경우 관리자가 지정한 문자열이나 사이트 혹은 서버 상에 존재하는 특정 파일을 표시해준다.
- ErrorDocument 403 http://dci.sppo.go.kr

(나) 로그 파일의 형식 및 로그 내용 분석

o access_log, error_log의 형태

o 클라이언트의 IP

o 클라이언트의 접속 시간 정보

o 클라이언트의 요청방식(GET, POST) 및 요청 내용(URL)

o 상태코드 정보

- 음영 처리된 부분은 HTTP/1.0에서 지원하는 코드이다.

(표 3-1) HTTP/1.1 상태코드

상태코드	설명	상태코드	설명
100	Continue	404	Not Found
101	Switching Protocols	405	Method Not Allowed
200	OK	406	Not Acceptable
201	Created	407	Proxy Authentication Require
202	Accepted	408	Request Time-out
203	Non-Authoritative Information	409	Conflict
204	No Content	410	Gone
205	Reset Content	411	Length Required
206	Partial Content	412	Precondition Failed
300	Multiple Choices	413	Request Entity Too Large
301	Moved Permanently	414	Request-URI Too Large
302	Moved Temporarily	415	Unsupported Media Type
303	See Other	500	Internal Server Error
304	Not Modified	501	Not Implemented
305	Use Proxy	502	Bad Gateway

400	Bad Request	503	Service Unavailable
401	Unauthorized	504	Gateway Time-out
402	Payment Required	505	HTTP Version not supported
403	Forbidden		

(다) 보안 관점의 로그 분석

o 홈페이지 취약점 공격시 로그 패턴 분석

- 특정 파일에 대한 연속적인 요청

o 웜에 의한 로그 패턴

- codered나 nimda 웜에 의한 로그 패턴
- default.ida?xxx

o 취약점 분석 프로그램에 의한 로그 패턴

- 짧은 시간 동안 많은 페이지가 요청될 경우
- 해당 서버와 관련 없는 파일들의 요청이 많을 경우

(라) 용도에 따른 별도 로그 저장 방법

o 환경 설정을 통한 특정 URL에 대한 로그 별도 저장(SenEnvIf)

o 리눅스 서버에서의 윈도우 웜에 의한 로그 기록 제외

- access_log에 codered나 nimda 웜에 의한 로그 제외

(2) IIS 로그 관리 및 분석

o 로그 파일의 위치 및 로그 관련 설정

o 로그 파일의 형식 및 로그 내용 분석

- Host, AuthUser, Time, Service, ServerIP, Status, Request, Filename 등

o 보안 관점의 로그 분석

- 홈페이지 취약점 공격시 로그 패턴 분석
- codered 공격시 IIS 로그 패턴 등

1.3.4 web 서비스 공격 유형[1급]

web 서비스에 있어서 공격유형은 크게 web 사용자 클라이언트의 취약점을 이용한 사용자 컴퓨터 공격과 웹 서버의 취약점을 이용한 웹서버 공격으로 나뉜다.
특히 웹서버에 대한 공격은 해당 네트웍의 방화벽의 필터링을 관통하여 내부네트웍

을 공격하는 시작지점이 되므로 특히 주의하여야 한다. 웹 서비스 관련 공격은 최신 해킹 경향에 따라 끊임없이 새로운 공격기법이 발견되므로, 항상 최신 보안 경향에 관심을 갖고 있어야 한다. 또한 매년 발표되는 OWASP TOP10에 대해 이해하고 있어야 한다.

o 핵심가이드

- 웹서버 버그 공격 기법
- 각종 공격기법의 원리 및 대응 방법 이해

(1) 웹서버 버그 공격

o 서버의 버전 정보가 갖는 의미
o 아파치 서버에 존재하는 최신 취약점 정보들
o IIS 서버에 존재하는 최신 취약점 정보들
  - IIS 서버 공격을 응용한 인터넷 웜 공격기법(codered, nimda 등)

(2) 부적절한 파라미터 조작

o HTML 요청을 변조하여 보안 매커니즘 우회
o 인수를 조작하여 시스템 명령어 실행
o Perl이나 Shell Script 등의 스크립트 기반 언어로 WK여진 웹 어플리케이션의 경우 일반 변수에 특정 문자열을 삽입할 경우 이를 적절히 처리하지 못하고 시스템의 명령어를 실행
o 취약성 판단 방법
  - tained 인자를 사용하고 있는지 않은지 확인
  - 소스 코드 상세 분석
  - OWASP의 WebScrab고 같은 도구를 사용하여 tained 인자 검사
  - http://www.owasp.org/software/webscarab.html
o 대응 방법
  - 데이터 유형 검증
  - 허용된 문자셋 검증
  - 최대/최소 길이 검증
  - Null 값의 허용 여부 검증
  - 반드시 필요한 인자와 그렇지 않은 인자 검증

- 중복 허용 여부 검증
- 숫자의 범위 검증
- 타당한 것으로 지정된 값/패턴 검증

(3) 원격지 파일의 명령 실행

o 게시판 소스 코드 중 include문을 이용하여 passthru나 system과 같이 원격에서 명령 실행이 가능한 함수를 추가하여 원격지에서 명령을 실행
o 취약성 판단 방법
  - 게시판 소스 코드에 include문이 있는지 확인
  - include문에 의해 원격지의 파일을 포함할 수 있는지 확인
o 대응 방법
  - PHP의 경우 php.ini 파일에서 allow_url_fopen 옵션을 Off 값으로 설정

(4) SQL Injection

o 공격자가 입력값을 조작하여 원하는 SQL 구문을 실행하는 기법
o 잠재적인 SQL 구문의 구조 확인 후 적절히 실행되는 문자들의 결합을 찾을 때까지 입력을 조작하는 기법
o 부적절한 입력값을 전달하여 에러를 발생시켜 SQL 구문을 확인하는 방법
o MS-SQL상에서의 시스템 명령어 실행
  - xp_cmdshell 저장 프로시져를 이용한 시스템 명령어 실행
o 취약성 판단 방법
  - 검색어 필드 및 로그인 필드에 큰따옴표, 작은따옴표, 세미콜론을 입력하여 DB 에러가 발생하는지 확인
  - 로그인 ID 필드에 ' or 1=1;--과 비밀번호 필드에 아무값이나 입력한다.
  - 로그인 ID 필드와 비밀번호 필드에 ' or '1'='1을 입력해 본다.
o 대응 방법
  - 사용자의 입력에 특수 문자가 포함되어 있는지 검증
  - SQL 서버의 에러 메시지 미표시
  - 일반 사용자 권한으로 시스템 저장 프로시져에 접근 불허

(5) File Upload

게시판에서 파일 업로드가 가능한 경우 해당 게시판의 웹 어플리케이션과 동일한 언어의 스트립트 파일을 업로드한 후에 다시 이를 SSI(Server Side Interpreter) 특성을 이용하여 실행시킴으로서 웹 서버의 내부 명령어를 실행하는 공격

o 취약성 판단 방법
  - 게시판에 글쓰기 권한과 파일 첨부 기능이 있는지 확인한 후 확장자가 jsp, php, asp, cgi 등의 파일들이 업로드가 가능한지 확인
o 대응 방법
  - 업로드 되는 파일의 확장자 검증
  - 업로드 파일을 위한 디렉토리에 실행 권한 제거

(6) File Download

다운로드를 위한 게시판의 파일을 이용하여 임의의 문자나 주요 파일명의 입력을 통해 웹 서버의 홈 디렉토리를 벗어나 시스템 내부의 다른 파일로 접근하여 다운로드하는 공격

o 취약성 판단 방법
  - 게시판에 특정 파일을 이용하여 파일을 다운로드 받는 페이지가 있는지 조사한 후 다운로드 파일명을 시스템의 중요한 파일의 위치와 이름으로 바꾼 후 다운로드 시도
o 대응 방법
  - 파일 다운로드시 절대경로 사용 대신 특정 파일을 이용하여 다운로드
  - 파일 다운로드시 지정된 파라미터 값이 모두 입력되었는지 검증

(7) 쿠키/세션 위조

o 클라이언트에 전달된 쿠키 분석
o 각 인증에 사용되는 웹 페이지 분석
o 매번 접속할 경우 변하는 쿠키 부분과 변하지 않는 쿠키부분 분석
o 쿠키의 이름을 보고 내용 유추
o GET 방식과 POST 방식을 이용하여 위조된 쿠키로 인증을 통과하는 기법

o 취약성 판단 방법

- 웹브라우저 주소창에 javascript:document.cookie; 를 입력하여 내용 확인 후 해당 세션 쿠키를 사용하는 웹 어플리케이션 소스 점검을 통해 불법 변조 탐지 루틴이 있는지 확인

o 대응 방법

- 전송 중의 자격 증명 보호
- cookie 대신 Server Side Session을 사용

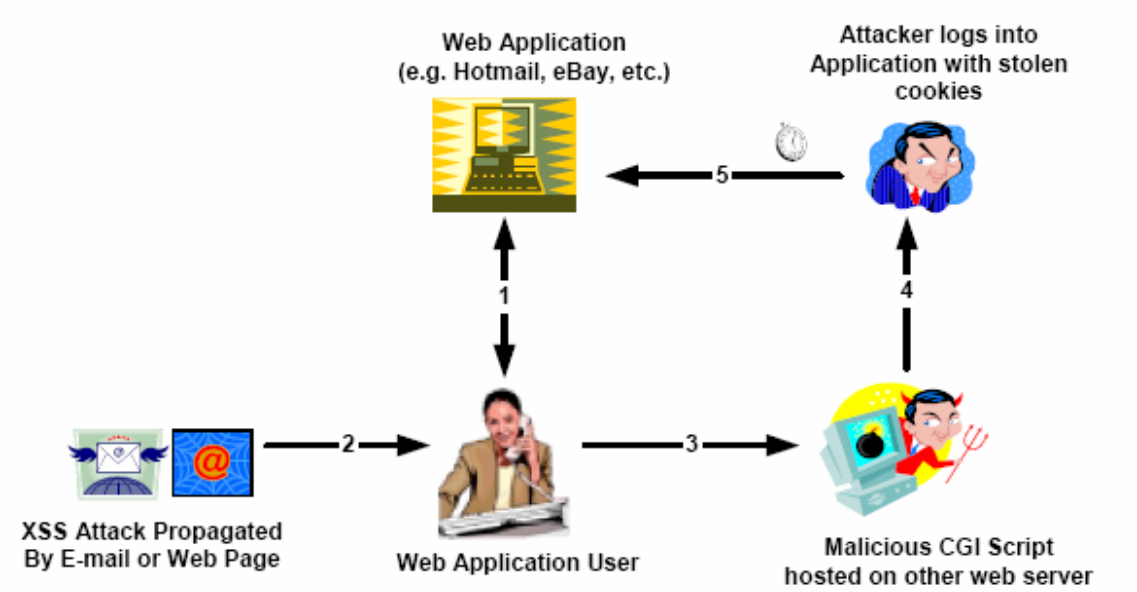
(8) Cross Site Scripting(XSS)

o 게시판의 글에 원본과 함께 악성코드를 삽입하여 글을 읽을 경우 악성코드가 실행되도록 하여 클라이언트의 정보를 유출하는 클라이언트에 대한 공격 기법

o 웹 페이지가 사용자에게 입력 받은 데이터를 필터링하지 않고 그대로 동적으로 생성된 웹 페이지에 포함하여 사용자에게 재전송할 때 발생

o 취약성 판단 방법

- 게시판에 글쓰기와 같이 단문 이상의 입력이 가능한 부분에 <script> 태그를 입력이 사용 가능할 경우 취약점 존재



(그림 3-5) 크로스 사이트 스크립팅 과정

o 대응 방법

- Server.HTMLEncode() 메소드 사용(ASP)
- htmlspecialchars() 함수 사용(PHP)
- strip_tags() 함수를 이용하여 문자열로부터 HTML 태그와 PHP 태크를 제거

라. OWASP TOP10

매년 OWASP에서 발표되는 가장 심각한 10가지 웹 어플리케이션 보안 취약점에 대해서 이해하는 것이 필요하다. [1,2급 공통출제].

o 핵심가이드

- OWASP에서 발표하는 웹에 대한 10가지 취약점 이해

(1) 입력값 검증 부재

웹 요청 정보가 웹애플리케이션에 의해 처리되기 이전에 적절한 검증이 이루어지고 있지 않다. 공격자는 이 취약점을 이용하여 웹 애플리케이션의 백엔드 컴포넌트를 공격할 수 있다.

(2) 취약한 접근 통제

인증된 사용자가 수행할 수있는 작업을 적절히 제한하지 않고 있다. 공격자는 이 취약점을 이용하여 다른 사용자의 계정에 접근하거나, 민감한 정보가 담긴 파일을 열람하거나, 허용되지 않은 작업을 수행할 수있다.

(3) 취약한 인증 및 세션 관리

계정 토큰과 세션 토큰이 적절히 보호되고 있지 않다. 공격자는 암호나 키, 세션 쿠키, 기타 인증 관련 토큰을 공격하여 인증을 우회하고 다른 사용자의 ID를 가장할 수 있다.

(4) 크로스 사이트 스크립팅(XSS) 취약점

웹 애플리케이션이 다른 사용자의 브라우저를 공격하는 도구로 사용될 수 있다. 공격이 성공하는 경우 일반 사용자의 세션 토큰이 노출되거나, 사용자의 컴퓨터를 공격하거나, 다른 사용자를 속이기 위해 위조된 컨텐츠를 보여주게 된다.

(5) 버퍼 오버플로우

웹 애플리케이션 컴포넌트가 사용자의 입력값을 적절히 점검하지 않는 언어로 작성되어 다운될 수 있다. 특수한 경우에는 공격자가 해당 프로세스의 권한을 획득할 수 있다. 이 컴포넌트로는 CGI, 라이브러리, 하드웨어 드라이버, 웹애플리케이션 서버 컴포넌트 등이 포함된다.

(6) 삽입 취약점

웹 애플리케이션이 외부 시스템이나 자체 OS에 접근할 때 입력받은 인자를 그대로 전달한다. 공격자가 해당 인자로 악의적인 명령어를 삽입하는 경우, 해당 외부 시스템은 웹애플리케이션으로 인해 입력받은 명령어를 실행할 수 있게 된다.

(7) 부적절한 에러 처리

일상적인 운용 과정 중에 발생하는 에러 상황에 대해 적절한 처리가 이루어지지 않는다. 공격자가 웹 애플리케이션이 처리하지 못하는 에러가 발생하도록 유도하여, 해당 시스템에 대한 상세 정보를 획득하거나, 서비스를 방해하거나, 보안 메커니즘이 작동하지 않도록 할 수 있으며, 서버가 다운될 수도 있다.

(8) 취약한 정보 저장방식

웹 애플리케이션은 정보나 인증 관련 토큰을 보호하기 위해 암호화를 자주 사용한다. 암호화 관련 기능이나 코드는 적절하게 구현하기가 어려움이 이미 증명되었으며, 많은 경우 오히려 보안상 바람직하지 않은 결과를 초래한다.

(9) 서비스 방해 공격

공격자가 다른 정당한 사용자가 사이트에 접속하거나, 애플리케이션을 사용하는 것을 방해하기 위해 웹 애플리케이션의 리소스를 고갈시킬 수 있다. 공격자는 또한 다른 사용자가 본인 소유의 계정을 사용하지 못하도록 계정을 잠글 수 있으며, 심지어 웹 애플리케이션 전체가 멈추도록 할 수 있다.

(10) 부적절한 환경 설정

강화된 서버 환경 설정 표준을 보유하는 것은 안전한 웹 애플리케이션에 있어 결정적으로 중요한 부분이다. 해당 서버는 보안에 영향을 미치는 다양한 환경 설정 옵션이 있으며, 벤더 출하시에는 기본적으로 안전하지 않은 상태로 출시된다.

1.3.5 웹보안 개발[1급]

웹 서버에 대한 공격은 정형화된 취약점을 정형화된 공격툴로 공격하는 것이 아니기 때문에 정형화된 점검툴로는 사이트별로 특징적으로 존재하는 취약점에 대하여 완전한 점검이 일반적으로 불가능하다. 따라서 각 사이트별로 웹 환경을 개발할 때에 보안에 입각하여 개발을 하는 것이 중요하다.

o 핵심가이드

- ASP, PHP, JSP 등의 보안 프로그래밍 중요성 정도 이해

(1) 웹 보안 프로그래밍

o 파일 업로드 공격 방지 개발 방법
o 파일 다운로드 공격 방지 개발 방법
o SQL Injection 공격 방지 개발 방법
o 쿠키/세션 위조 공격 방지 개발 방법

1.3.6 XML 기반 Web 보안[1급]

XML은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다. 이는 인터넷에서 기존에 사용하던 HTML의 한계를 극복하고 SGML의 복잡함을 해결하는 방안으로써 HTML에 사용자가 새로운 태그(tag)를 정의할 수 있는 기능이 추가되었다. 또한 XML은 SGML의 실용적인 기능만을 모은 부분집합(subset)이라 할 수 있으며, 어떤 플랫폼에서나 읽을 수 있는 포맷을 제공하는 범용성을 갖는다.

(1) UDDI 개념과 특징

o UDDI의 개념
- UDDI(Universal Description, Discovery and Integration)
- 인터넷에서 전 세계 비즈니스 목록에 자신을 등재하기 위한 XML 기반의 레지스트리이다.

- 월드와이드웹(WWW)에서 상호 온라인 거래를 원활히 하고 전자상거래의 상호 운용을 하기 위한 것으로 비즈니스 이름, 제품, 위치 혹은 웹서비스(web service) 등으로 목록을 작성하여 사용자에게 제공하는 것이다.
- UDDI를 통하여 기업은 인터넷에서 어떤 거래를 원하는지 공개할 수 있고, 기업간 전사상거래의 발전이 촉진된다. UDDI는 기업과 그 기업이 제공하는 서비스에 관한 정보를 공개하거나 찾아내고 종합하기 위해 오픈된 아키텍처를 제공함으로써 모든 기업이 혜택을 받을 수 있다.

o UDDI의 특징
- General 메타 데이터를 제공
- 벤더의 보장(commitment) 제공
- 여러 방식의 프로토콜이 적용 가능
- 개방성이 없음
- RDB(Relational Database Management) 쿼리 인터페이스가 없음

(2) SOAP의 개념과 특징

o SOAP 개념
- SOAP(Simple Object Access Protocol)
- 1999년 MS에 의해 발표되었고 2000년 5월 W3C에 의해 표준으로 채택(SOAP1.1)
- XML과 HTTP 통신을 기반으로 네트워크상에 존재하는 각종 시스템간의 호출을 효율적으로 실현하기 위한 방법으로 제시하는 통신규약

o SOAP 특징
- 프로토콜과 언어 독립성을 가지고 있다
- 플랫폼 및 운영체제로부터 독립성을 가지고 있다.
- SOAP XML 메시지 첨부를 지원(다중 MIME 구조 사용)

(3) WSDL의 개념과 특징

o WSDL의 개념
- WSDL(Web Service Description Language)
- WSDL은 특정 비즈니스가 제공하는 서비스를 설명하고, 개인이나 다른 회사들이

그러한 서비스에 전자적으로 접근할 수 있는 방법을 제공하기 위해 사용되는 XML 기반의 언어이다.
- WSDL은 마이크로소프트, IBM 및 Ariba 등에 의해 주도된 UDDI의 기본이라 할 수 있다. 즉, UDDI는 기업들이 자신들의 서비스 내용을 인터넷 상에 스스로 등록할 수 있게 해주는 XML 기반의 등록처이며, WSDL은 그렇게 하기 위한 언어이다. WSDL은 마이크로소프트의 SOAP와 IBM의 NASSL로부터 파생되었지만, 이제 UDDI 등록처에 비즈니스 서비스를 명시하는 수단으로서, NASSL과 SOAP 둘 모두를 대체할 수 있다.

o WSDL 특징
 - 추상적인 구문과 메시지
 - WSDL 파일에는 포트 정보를 제공할 필요가 없음
 - WSDL 스키마는 상위 레벨 혹은 언어상의 주된 요소(element)를 정의한다.

(4) XML 보안

o XML 전자서명
 - XML 전자서명에 대한 구문과 처리 규칙
 - 무결성, 메시지 및 서명자 인증과 부인방지
 - 어떤 디지털 컨텐츠에도 적용 가능
o XML Encryption
 - XML 암호에 대한 구문과 처리 규칙, 수행규칙
 - 3DES, RSA-v1.5, base64 알고리즘 지원
 - XMS 암호의 적용 대상(일반 XML 문서, 일반 2진 형식의 데이터)
o XKMS(XML Key Management Specification): PKI 서비스 프록시
o SAML(Security Assertion Markup Language): 인증, 속성, 승인 Assertion
o XACML(eXtensible Markup Language): XML 기반의 접근 제어
o WS-Security(Web Service Security)

1.4 DB 보안

정보조직에서 가장 중요한 정보 자산 가치를 갖는 것이 바로 데이터베이스이다. 현실 세계에서의 정보 보안의 근본적인 보호대상이 바로 데이터베이스에 수록되어

있는 중요 정보 자산이라 할 수 있을 것이다. 이러한 데이터베이스 보안에는 크게 DB 데이터 보안과 DB 관리자 권한 보안으로 나뉠 수 있으며, 이를 토대로 효율적인 DBMS 보안 운영이 가능할 것이다.

1.4.1 DB 데이터 보안

본 영역은 데이터베이스의 기본개념을 기반으로 DB 데이터 보안에 관한 내용의 이해를 요구한다.

가. 데이터베이스 기본 개념

o 핵심가이드

- DB 관련 기본 용어의 이해 : 정보, 데이터, 데이터베이스, DBMS, 키 등
- 키의 종류와 특성의 이해.

(1) 정보와 데이터

o 데이터 : 관찰이나 측정을 통해서 수집된 사실이나 값으로 수치, 스트링 등의 형태로 표현

o 정보 : 데이터가 가공된 형태로, 의사 결정을 할 수 있게 하는 데이터의 유효한 해석이나 상호관계, 의미를 나타냄

(2) 데이터베이스의 정의

o 데이터베이스 : 한 조직의 여러 응용 시스템이 공용하기 위해 최소의 중복으로 통합, 저장된 운영 데이터의 집합

(3) 데이터의 종류

o 통합된 데이터 (Integrated Data) : 최소의 중복 / 통제된 중복

o 저장 데이터 (Stored Data) : 컴퓨터가 접근 가능한 매체에 저장

o 운영 데이터 (Operational Data) : 조직의 운영에 꼭 필요한 필수적인 데이터

o 공유 데이터 (Shared Data) : 여러 응용 프로그램이 공동으로 허용

(4) DBMS 의 개념

o DBMS : 응용 프로그램과 데이터의 중재자로서 모든 응용 프로그램들이 데이터베이스를 공용할 수 있게끔 관리해 주는 소프트웨어

(5) 관계형 데이터 모델의 구성 요소

o 릴레이션 (Relation) : 열과 행으로 이루어진 테이블

o 속성 (Attribute) : 테이블 ㅅ혹의 데이터에 대한 고유한 특성을 나타내는 테이블의 열

o 투플 (Tuple) : 테이블의 각 행

o 도메인 (Domain) : 한 릴레이션에서 특정 속성이 가질 수 있는 데이터 형식을 지닌 모든 가능한 값의 집합

o 키 (Key) : 투플을 구분시켜 주는 하나 또는 그 이상의 속성들의 모임

o 무결성 제약 조건 : 데이터 무결성을 보장하기 위해 데이터에 적용되는 일련의 규칙

(6) 키 (Key)

o 릴레이션의 투플을 유일하게 식별할 수 있는 속성의 집합을 말함

(가) 키의 특성

o 유일성

- 속성의 집합인 키의 내용이 릴레이션 내에서 유일하다는 특성
- 릴레이션 내에서 중복되는 투플이 존재하지 않는 것

o 최소성

- 속성의 집합인 키가 릴레이션의 모든 투플을 유일하게 식별하기 위해 꼭 필요한 속성들로 구성된 것을 의미
- 속성들의 집합에서 특정 속성 하나를 제거하면 투플을 유일하게 식별할 수 없는 경우에 해당

(나) 키의 유형

o 후보키 (Candidate Key)

- 키의 특성인 유일성과 최소성을 만족하는 키를 지칭
- 예) <학번>, <이름, 학과>

o 슈퍼키 (Super Key)
- 유일성은 있으나 최소성이 없는 키를 지칭
- 특정 속성을 제거하면 투플을 유일하게 식별하지 못하는 것
- 예) <이름, 학과, 학년>

o 기본키 (Primary Key)
- 여러 개의 후보키 중에서 하나를 선정하여 사용하는 것을 지칭
- 예) <학번>, <이름, 학과> 후보키 중에서 하나를 선정하는 것

o 대체키 (Alternate Key)
- 여러 개의 후보키 중에서 기본키로 선정되고 남은 나머지 키를 지칭
- 기본키를 대체할 수 있는 키라는 의미
- 기본키를 <학번>으로 선정했다면, <이름, 학과>를 지칭

o 외래키 (Foreign Key)
- 어느 한 릴레이션 속성의 집합이 다른 릴레이션에서 기본키로 이용되는 키를 지칭

나. DB 데이터 보안

o 핵심가이드
- DB 보안 요구 사항의 이해
- 데이터 무결성의 특징 이해
- 접근 통제, 추론 통제, 흐름 통제 등 DB 보안 통제 이해
- DB 위협 종류에 대한 숙지 : 우연적 위협, 의도적 위협 등
- DB 보안 위협에 따른 결과 유형 이해
- 접근 제한, 허가 규칙, 가상테이블, 암호화 등의 DB 보안 방법 이해

데이터베이스 보안은 데이터베이스에 저장되어 있는 데이터에 대한 허가 받지 않은 접근, 의도적인 데이터 변경과 파괴, 그리고 데이터의 일관성을 저해하는 우발적인사고 등으로부터 데이터 혹은 데이터베이스를 보호하는 일련의 활동을 말한다. 데이터보안에서는 크게 DB 보안 요구사항, DB 보안 통제, DB백업과 복구로 나눌 수 있다.

(1) DB 보안 요구 사항

o 부적절한 접근 방지
o 추론 방지
o 데이터 무결성
  - 데이터베이스 내에 있는 자료 값들이 정확하도록 보장하는 관리 작업
  - 잘못된 갱신으로부터의 보호나 불법적인 조작에 대한 보호를 통한 정확성 유지
  - 보한 시스템을 통하여 DMBS의 무결성 서브 시스템의 지원을 받아 제어
o 감사 기능
o 사용자 인증

(2) DB 보안 통제

o 접근 통제
  - 데이터베이스는 사용자가 가진 접근 권한에 따라서 논리적으로 분리
  - 데이터베이스 관리자(Database Administrator : DBA)는 누가 어떤 부분의 데이터에 접근 가능하며, 어느 수준까지(예. 필드 수준인지 레코드 수준인지) 접근가능한지, 그리고 어느 기능까지만 (예. 읽기/쓰기/변경/삭제/추가 등의 기능) 허락할 것인지를 결정
  - 이와 같은 체계적인 정책을 확립하고 수행함으로써 데이터베이스보안을 유지하는 것.

o 추론 통제
  - 간접적으로 노출된 데이터 노출을 통해 다른 데이터를 추론하여 다른 데이터가 공개되는 것을 방지하는 것
  - 통계적인 자료에서 많이 발생
  - 완벽한 해결책은 존재하지 않음
  - 추론 방지를 위한 세 가지 방법
    · 허용 가능한 질의 제한
    · 질의의 응답으로 제공되는 데이터 한정
    · 데이터를 숫자의 경우 반올림하거나 일관성이 없는 질의 결과를 제공

o 흐름 통제
- 접근 가능한 객체들간의 정보 흐름을 조정
- 정보가 명시적으로 또는 암시적으로 보다 낮은 보호 수준의 객체로 이동하는 것을 검사
- 정보가 보다 낮은 수준의 보호 객체로 이동시 사용자는 보안 위협이 발생하게 되므로 통제 메커니즘에서 이를 거부하여 보안 유지

(3) DB 보안 위협

(가) 위협의 종류
o 우연적 위협
- 천재지변, 우발재해 : 시스템 하드웨어나 저장 데이터 손상
- 하드웨어나 소프트웨어 오류 및 버그 : 데이터의 비권한 접근, 권한 있는 사용자의 접근 거부
- 인간 오류 : 올바르지 못한 입력 등 비의도적 위반

o 의도적 위협
- 권한 사용자에 의한 위협 : 자신의 권한을 남용
- 적대적 행위자에 의한 행위 : 바이러스, 트로이 목마, 트랩도어

(나) DB 보안 위협에 의한 결과 유형
o 정보의 부당한 유출
o 데이터의 부당한 수정
o Aggregation
- 낮은 보안 증급의 정보 조각을 조합하여 높은 등급의 정보를 알아내는 것
- 개별 데이터 항목 보다 종합 정보의 보안 등급이 높은 경우 심각한 문제
- 각 지사의 영업 실적을 조합하여 대외비인 회사의 총 매출액 산정
o Inference
- 보안 등급이 없는 일반 사용자가 보안으로 분류되지 않은 정보에 정당하게 접근하여 기밀 정보를 유추해 내는 행위
- 보안 대책 : Polyinstantiation

(4) 데이터베이스의 보안의 유형

(가) 접근 제어 (Access Control)

o 허가 받지 않은 사용자의 데이터베이스 자체에 대한 접근을 방지하는 것

o 계정 및 암호

o DB에 발생한 조작에 대한 주체에 대해 트랜잭션 로그로 제공

(나) 허가 규칙 (Authorization Rules)

o 정당한 절차를 통해 DBMS 내로 들어온 사용자라 하더라도, 허가 받지 않은 데이터에 접근하는 것을 방지하기 위한 것

(다) 가상 테이블 (Views)

o 가상 테이블을 이용하여 전체 데이터베이스 중 자신이 허가 받은 사용자 관점만 볼 수 있도록 한정하는 것

(라) 암호화 (Encryption)

o 데이터를 암호화 하여 불법적으로 데이터에 접근하더라도 알 수 없는 형태로 변형 시키는 것

(5) DB 백업과 복구

o 장애유형

- 사용자 오류
- 명령문 장애 : 유효한 SQL 구성이 아닌 경우와 같이 명령문을 처리할 때 발생하는 논리적 상태. 명령문에 장애가 발생하면 명령문의 실행 결과가 자동으로 무효화되고 제어가 사용자에게로 돌아감
- 프로세스 장애 : 비정상적인 접속 해제나 프로세스 종료와 같이 DB를 액세스하는 사용자 프로세스에 발생하는 장애.
- 인스턴스 실패 : 인스턴스 (시스템 글로벌 영역과 백그라운드 프로세스)가 계속 작업을 수행할 수 없게 되었을 때 발생함.
- 매체고장

o 복구에 사용 되는 기술

- redo 로그
- 제어 파일
- 롤백

1.4.2 DB 관리자 권한 보안

o 핵심가이드
- 운영체제에서 제공하는 인증 방법 이해
- Kerberos 등의 네트워크 인증방법 이해

데이터베이스 관리자들은 특별한 operations을 수행해야 하므로 데이터베이스 관리자 username은 보다 안전한 인증 schema을 필요로 한다.

(1) Operation system에 의한 인증

(2) Network 인증 서비스 (kerberos 등)에 의한 인증

1.4.3 DBMS 운영

o 핵심가이드
- SSO, 인증, PKI, LDAP, OID 등 Oracle 운영 시 보안 설정 방법 이해
- Informix 서버 환경설정, 로그, 모니터링, 백업 및 복구 절차 이해
- MySql 보안 운영 방법 이해
- Mssql 보안 운영 방법 이해

실제의 DBMS 시스템을 운용함에 있어서 각 DBMS 플랫폼별로 주어진 보안 기능을 적절히 활용하여 안정성 있는 DB보안 관리를 해야 한다.

(1) Oracle 보안 운영

(가) Single Sign On 기능 설정

o SSO Server
- 사용자 인증 담당

- 클라이언트 브라우저에 저장한 쿠키 사용

o 파트너 어플리케이션

- SSO 프레임워크 내부에서 동작하는 애플리케이션
- 암호를 관리할 필요가 없어 사용자 관리 작업의 단순화
- 배포 비용과 관리 비용 절감

o MOD_OSSO

- Oracle HTTP Server가 SSO 파트너 애플리케이션이 될 수 있도록 해주는 확장

(나) 인증

(다) LDAP 응용

(라) PKI 응용

(마) OID (Oracle Internet Directory)를 통한 디렉토리 활용 보안

(2) Informix 보안 운영

(가) 서버 설정

o 환경 변수

o 디스크 파라메터 설정

o 공유 메모리 파라메터 설정

o Performance Tuning 파라메터 설정

(나) 디스크 관리

1) 용어

o Chunk : 물리적인 디스크 영역의 단위

o Page : 입출력의 기본 단위

o Extent : 테이블을 구성하는 물리적으로 연속적인 디스크 공간

o tblspace : 한 테이블을 구성하는 extent들의 논리적인 집합

2) extent를 구성하는 페이지

o Bitmap Page : extent안의 다른 페이지에 대한 정보를 갖는 페이지

o Data Page : 테이블의 데이터 행을 저장

o Remainder Page : page크기보다 큰 행을 저장할 때 남은 부분 저장

o Blob Page : blob 데이터 저장
o Free Page : extent에 할당되었으나 사용하지 않은 페이지
o Index page : index데이터를 저장하는 페이지

3) extent growth
o concatenation – 물리적으로 연속된 extent는 기존 extent에 추가
o doubling – 16의 배수번째 extent부터 기존 크기의 2배로 할당
o manual modification – alter table 구문으로 next extent 크기 조정

(다) 로그 관리

o 로그 모니터링 : onstat -l
o logical log 추가 : onparams -a -d dbspace이름 [-s 로그크기]
o logical log 삭제 : onparams -d -l 로그ID
o 로그 전환 : onmode -l
o physical log 크기 및 위치 변경 :
onparams -p -d dbspace이름 -s 로그크기 [-y]
o 데이터베이스 로깅 모드 변경 :
ontape -s { -N | -B | -U | -A } 데이터베이스이름
- N : no-logging
- B : buffered logging
- U : un-buffered logging
- A : ANSI mode logging

(라) 서버 모니터링

1) 모니터링 대상
o 메시지 로그
o 공유 메모리
o 논리 로그 공간 및 chunk 사용률
o 사용자 활동
o 서버 리소스 사용률
o 잠금 사용률

2) 모니터링 도구

o SMI(System Monitoring Interface)
  - sysmaster 데이터베이스
  - SQL을 이용한 서버 상태 확인
o onstat 유틸리티
  - 서버 메모리에 대한 모니터링
  - 디스크 입출력 불필요
  - lock을 사용하지 않으므로 서버 성능이 저하되지 않음
o oncheck 유틸리티
  - 디스크의 인덱스나 데이터 페이지 점검
  - 깨진 인덱스 복구
  - 디스크의 구조 검사 및 디스크 정보 출력

(마) 백업 및 복구

1) 백업

o 백업 툴
  - ontape
  - on-bar
o Incremental 백업
  - 0 레벨 / 1 레벨 / 2 레벨
o 백업 단계
  - 공간 확인 : Logical log의 반 이상 여유 공간 필요
  - full checkpoint를 실행
  - 백업 받을 chunk의 페이지 리스트 작성
  - physical log를 위한 임시 테이블 생성
  - 백업 thread 작동
o 서버 관리 작업 후 백업 과정으로 서버 상태 동기화
  - 미러(mirror) 추가
  - Logical Log 추가, 삭제
  - Physical Log의 크기나 위치 변경
  - Chunk 또는 dbspace 추가, 삭제

- Storage manager 구성 변경

2) 복구

o 복구 프로세스
  - physical restore
  - logical restore

o 복구 시나리오
  - cold restore
  - warm restore
  - mixed restore

o Log Salvage
  - 서버가 off-line일 때 restore하기 전 아직 백업 받지 못한 logical log를 백업하는 행위
  - cold restore 단계에서 자동 또는 수동으로 수행
  - on-line상태에서 백업 받은 log와 마찬가지로 logical restore
  - 데이터 손실을 최소화

(3) Mysql 보안 운영

(가) 서버의 기동과 종료

o 명령 라인 상에서 수동으로 서버 프로그램을 기동하고 종료하는 법 숙지
o 원하는 시각에 자동으로 서버 프로그램 기동 및 종료 스크립트 작성법 숙지
o 서버의 폭주 및 정상적으로 시작되지 않는 경우 대처 방법 숙지

(나) 사용자 계정 유지관리

o MySQL의 사용자 계정과 UNIX나 Windows의 로그인 계정 간의 차이점 숙지
o 계정의 연결 가능한 서버 지정
o 패스워드 재설정법 숙지

(다) 로그 파일의 유지

o 로그 파일의 형태
o 로그의 교체와 기간 만료일 지정
o 사용자의 파일 시스템이 로그로 꽉 차버리게 만드는 것 방지

(라) 데이터베이스 백업과 복사

- o 시스템에 심각한 장애가 생겼을 경우를 대비
- o mysqldump : 서버를 중지시키지 않고 백업 파일 생성 가능.
- o 디스크가 용량 찼을 때 백업 후 데이터베이스 이관 필요

(마) 데이터베이스 리플리케이션

- o 백업하거나 복사
- o 데이터베이스의 상태를 그대로 보존

(바) 서버 설정 및 튜닝

- o 최상의 상태에서 서버가 수행되도록 하는 작업
- o 더 많은 메모리 확보
- o 빠른 디스크 사용
- o 효과적인 쿼리와 매개변수 사용

(사) 복수의 서버

- o 테스트 서버 용도
- o 데이터 보안에 대한 보장

(아) MySQL 소프트웨어의 업데이트

(자) 파일 시스템 보안

(차) 서버 보안

- o 적절한 권한을 부여

(카) 고장 복구

(파) 예방 조치

(4) MS sql 보안 운영

(가) 인증 모드

MS sql 서버는 보안 액세스를 위한 두 가지 인증 모드 제공

1) Windows 인증 모드

o SQL Server 기본 인증 모드
o DB의 인증 절차를 Windows 사용자 인증 방법 사용
o Windows 사용자 또는 그룹에 따라 SQL Server에 대한 액세스 부여
o 트러스트 연결
o 데이터베이스 관리자가 사용자에게 접근 권한 부여 가능
o 윈도우즈 인증 로그온 추적 시 SID 값 사용

2) 혼합 모드
o Windows 인증이나 SQL Server 인증을 사용
o SQL Server로 인증된 사용자의 사용자 이름과 암호 쌍은 SQL Server 내에 유지
o 표준 Windows 로그온을 사용할 수 없는 경우에 이용
o 트러스트되지 않은 연결 (SQL 연결)

(나) MS sql 에서의 역할

o 역할 : 조직에서 작업자 그룹이 수행하는 작업
o 역할에 사용 권한을 부여 가능

1) Public 역할
o 데이터베이스 사용자에게 기본 사용 권한을 제공
o 삭제 불가
o 모든 데이터베이스 사용자는 이 역할의 구성원이 됨

2) 미리 정의된 역할
o 미리 정의된 암시적인 사용 권한
o 다른 사용자 계정에 부여 불가

3) 고정 서버 역할
o 서버 범위
o 데이터베이스 외부에 존재
o Sysadmin : SQL Server에서 모든 작업을 수행
o Serveradmin : 서버 차원의 구성 옵션을 구성하고 서버를 종료

o Setupadmin : 연결된 서버를 추가/제거하고 일부 시스템 저장 프로시저를 실행
o securityadmin : 연결된 서버를 포함한 서버 차원 보안 설정과 CREATE DATABASE 사용 권한관리. SQL Server 인증 로그인에 대한 암호 재설정
o processadmin : SQL Server에서 실행 중인 프로세스를 종료.
o dbcreator : 모든 데이터베이스를 생성, 변경, 삭제, 복구.
o diskadmin : 디스크 및 파일 관리
o Bulkadmin : sysadmin 이외의 사용자가 bulkadmin 문을 실행하도록 허용

4) 고정 데이터베이스 역할
o 데이터베이스 수준에서 정의
o db_owner : 데이터베이스에서 모든 유지 관리 및 구성 작업 수행
o db_accessadmin : Windows 사용자, 그룹 및 SQL Server 로그인에 대한 액세스 권한 추가, 제거
o db_datareader : 모든 사용자 테이블의 모든 데이터 읽기 가능
o db_datawriter : 모든 사용자 테이블에서 데이터를 추가, 삭제 또는 변경 가능
o db_ddladmin : 데이터베이스에서 모든 데이터 정의 언어(DDL) 명령 실행 가능
o db_securityadmin : 역할 구성원 자격을 수정하고 사용 권한을 관리
o db_backupoperator : 데이터베이스 백업 수행
o db_denydatareader : 데이터베이스에 있는 사용자 테이블의 모든 데이터를 읽기 불가
o db_denydatawriter : 모든 테이블이나 뷰의 데이터를 추가, 수정 또는 삭제 불가

5) 사용자 정의 역할
6) 응용 프로그램 역할

(다) MS sql 서버 구축 보안 체크리스트

1) 설치전 환경 설정
o 물리적 보안
o 서버와 인터넷 사이에 방화벽 배치
o 서비스 격리

o SQL Server 서비스를 실행하는 최소한의 권한을 가진 Windows 계정 생성
o NTFS 파일시스템 사용

2) 설치
o 최신 서비스 팩과 보안 패치 설치
o 서비스 계정 : 최소한의 권한으로 SQL Server 서비스 실행
o 인증모드 : SQL Server 연결 시 Windows 인증이 필요
o sa 계정에 항상 강력한 암호 할당

3) 구성 옵션 및 설치 후 설정
o 이전 설정 파일 삭제 또는 보호
o 명명된 인스턴스를 위한 고정 포트 선택
o 로그인 감사 수준 설정
o 보안 감사 활성화
o sa 보안 : sa 계정에 강력한 암호 할당
o 예제 데이터베이스 제거

4) 보안 작업
o 정기적으로 모든 데이터를 백업, 복사본을 조직 외부의 안전한 위치에 보관
o 노출 및 기능 최소화
o 관리자 최소화
o 모든 SQL Server 계정에 복잡한 암호 사용
o 사용하지 않는 경우 데이터베이스 간 소유권 체인을 사용할 수 없도록 설정
o sysadmin 역할의 구성원만 xp_cmdshell을 실행할 수 있도록 설정
o 암호화 인증서를 설치하여 SSL 연결 사용
o 사용자를 SQL Server 역할 및 Windows 그룹으로 모아 사용 권한 관리 단순화
o public 데이터베이스 역할에 사용 권한을 허가 금지
o guest 계정 사용 금지
o 서버에 암호화된 연결 (SSL 또는 IPSEC) 사용
o SQL 삽입 방지

5) 권장하는 주기적 관리 절차
o MBSA(Microsoft Baseline Security Analyzer) 주기적 취약점 점금

o 로그인 스캔 하여 null password 사용 제거 및 강력한 암호 사용 권장
o 사용하지 않는 계정 삭제
o AutoStart로 표시된 저장 프로시저의 보안을 확인

1.4.4 DB 보안 개발[1급]

o 핵심가이드
- sql 인젝션 공격 방지 개발 기법 이해

데이터베이스에 접근 할 수 있는 데이터베이스 어플리케이션에 보안상 문제점이 있을 경우 이를 이용하여 DB 정보에 주어진 권한을 넘어 접근할 수가 있게 된다. 특히 최근엔 웹서버와 데이터베이스 서버 간에 연동이 되어 정보서비스를 구축하는 웹-DB 연동 어플리케이션 개발 과정에서 보안을 고려한 개발이 더욱 요구된다.

(1) DB 어플리케이션 보안 프로그래밍

(가) 웹을 통한 sql injection 공격 방지 개발 방법

o 원시 ODBC 에러를 사용자가 볼 수 없도록 코딩
o 데이터베이스 어플리케이션의 최소 권한으로의 구동
o 데이터베이스 내장 프로시져 사용
o 테이블 이름, 컬럼 이름, sql 구조 등이 외부 HTML에 포함되어 나타나서는 안 됨

2. 전자상거래 보안

전자상거래는 소비자와 기업이 컴퓨터라는 매체를 통하여 이루어지는 거래이기 때문에 일반 상거래와는 달리 신원확인의 어려움, 대금결제, 프라이버시 침해, 그리고 전자상거래와 관련한 보안문제들이 발생할 수 있다. 전자상거래에 있어서 보안문제는 수많은 고객들이 인터넷을 이용한 전자상거래를 하고자 하는 경우에 많은 제약으로 작용하기 때문에 이러한 보안상의 문제가 중요한 문제로 부각되고 있다.

2.1 전자상거래 기술

인터넷의 상업적 이용이 급증하면서 수많은 기업들은 인터넷을 단순히 정보만을 제공하는 것이 아니라 온라인으로 제품을 구입하고 판매하는 등 기업경영전반에 걸쳐 적극적으로 활용하고 있다. 기업경영전반에 걸친 인터넷의 활용은 네트워크보안 및 거래보안과 관련한 제반 문제가 발생됨으로써 정보에 대한 기밀성, 무결성 그리고 인증 기능 등의 보안기술에 대한 중요성이 크게 증가하고 있다. 이 절에서는 안전한 월드와이드 웹 전자상거래를 구현하기 위한 방법들에 대해 이해하고 있는가를 평가한다.

2.1.1 암호시스템

암호는 정보를 보호하는데 중요한 수단을 제공하며 컴퓨터 보안의 많은 부분에서 사용된다. 고대에는 단순한 자료보호를 위하여 사용되던 암호가 현대로 접어들면서 다양한 보안서비스를 제공할 뿐 아니라 정보보호 및 전자상거래 문제를 해결하는데 이용되고 있다. 암호화의 목적은 전송되는 정보가 전달하고자 하는 상대방 이외의 다른 사람한테 읽히거나 내용이 파악되는 것을 막기 위한 것이다.

본 절은 정보보호론의 암호이론 부분과 중복되는 내용이 많으므로 아래와 같이 정보보호론의 암호학 출제영역을 참조하기 바란다.

o 관련내용 : 정보보호론의 1.암호학 참조

2.1.2 전자서명

정보를 암호화하여 상대편에게 전송하면 부당한 사용자로부터 도청을 막을 수는 있다. 하지만, 그 전송 데이터의 위조나 변조 그리고 부인 등을 막을 수는 없다. 이러한 문제점들을 방지하고자 디지털 서명이 등장하게 되었다.

본 절에서는 디지털 서명의 일반적인 특징 및 기능에 대해 평가한다. 다만 정보보호론의 암호학 출제영역과 중복되는 내용이 많으므로 전자서명의 기술적인 부분은 아래를 참조하기 바란다.

o 관련내용 : 정보보호론의 1.암호학 참조

2.2 전자상거래 프로토콜

기존의 대표적인 지불 프로토콜인 최종 개체간의 안전한 채널을 제공하기 위한 SSL 프로토콜, 안전한 전자지불을 보장하는 SET 프로토콜, 그리고 전자화폐 프로토콜들의 동작 원리와 프로토콜의 구성, 그리고 실현을 위한 요소 기술 등을 중심으로 이해하고 있는가를 평가한다.

o 인터넷이 활발히 구축되면서 사이버 공간상의 전자상거래가 활발히 추진되고 있으며 전자상거래가 웹에서 이루어지는 방법들은 반드시 그 자체의 안전성이 보장되어야 한다. 현 시점에서, 전자상거래 보안은 지불 방식에 중점을 두고 있으며, 다양한 보안 지불 프로토콜이 제안되고 있다.

2.2.1 전자 지불/화폐 프로토콜

o 핵심가이드

- 전자 지불 시스템의 분류 및 특징 이해
- Secure 전자 지불 서비스 모델 특징 이해
- 전자 화폐의 개념 및 특징, 그리고 문제점 이해
- 전자 상거래 보안 프로토콜 종류(S-HTTP,SSL) 및 특징 이해
- 전자 상거래 지불 프로토콜 종류(SET) 및 특징 이해
- 전자 지불 시스템 기술 요건 이해
- 전자 지불 시스템의 위협 요소 이해
- 안전한 전자지불 서비스를 위한 보안 메커니즘 이해

(1) 전자 지불 시스템의 분류

o 신용카드 기반 전자지불 시스템
- 보안 프로토콜
- 지불 프로토콜

o 전자화폐 기반 전자지불 시스템
- 네트워크형 프로토콜
- 가치저장형 프로토콜

o 전자수표 기반 전자지불 시스템
 - 전자수표 프로토콜

(2) Secure 전자지불 서비스 모델

지불 브로커 없이 독립적인 신용 구조를 가지고 현금과 유사한 개념의 전자적 지불 수단

o IC 카드형 전자화폐 시스템
  - IC 카드에 화폐 가치를 저장하여 지급수단으로 사용
  - 실세계의 거래에서 활용
  - 인터넷 상에서 적용은 연구 중
o 네트워크형 전자화폐 시스템
  - 네트워크 상에서 화폐가치 전송
  - 소액 거래시 유용

(3) 전자화폐

o 전자화폐의 개념 및 특성
o 전자화폐의 분류
  - IC 카드 형
  - Network 형
o 전자 화폐의 문제점
o 전자 화폐의 안전성 요구사항
  - 익명성
  - 오프라인성
  - 양도성
  - 분할성
  - 독립성 (완전 정보화)
  - 복사 및 위조 방지
  - 익명성 취소 기능

(4) 전자상거래 보안 프로토콜

o 보안프로토콜 기반의 상거래 절차

o 보안프로토콜의 종류

- S-HTTP(Secure-HTTP), SSL(Secure Socket Layer)

o 보안프로토콜 특징 및 암호 알고리즘

(가) S-HTTP(Secure-HTTP)

o 배경 : 1994년 EIT, NCSA, RSA에 의해 HTTP의 안전성 확보를 위해 개발

o 특징 : 응용계층에서 적용되며, 다양한 표준(PGP,PEM 등) 포용

·현황 및 전망 : 현재까지는 활성화되지 않고 있음

(나) SSL(Secure Socket Layer)

o ·배경 : 1993년 웹 서버와 브라우저간의 안전한 통신을 위해 넷스케이프社에 의해 개발

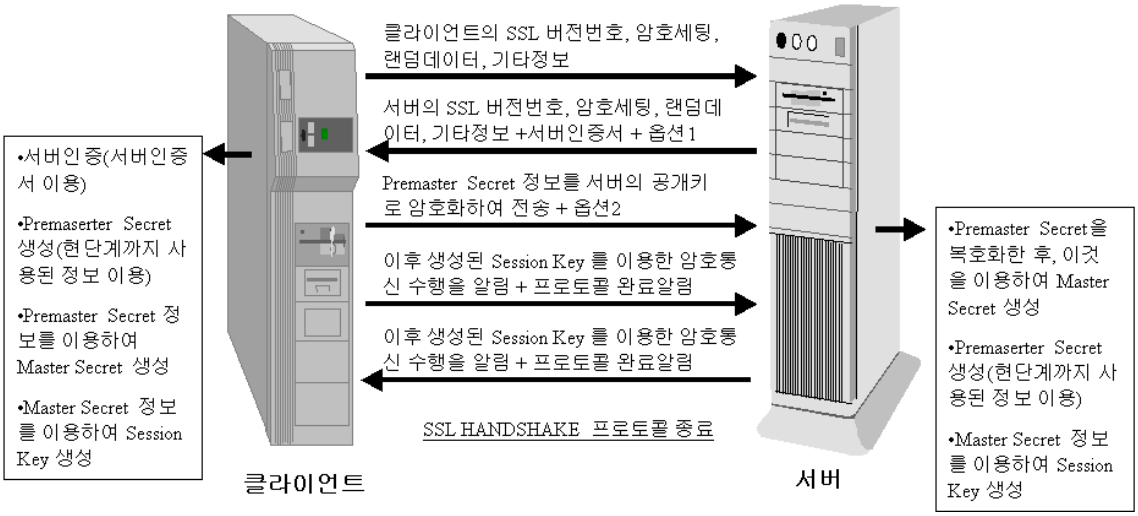
o ·특징 : 세션계층에서 적용되며, 응용계층의 FTP, TELNET, HTTP 등의 프로토콜의 안전성 보장

o SSL 프로토콜은 TCP/IP상의 어떤 포트를 사용해도 무방하나, 패킷 필터링 침입차단 시스템의 안전한 동작을 위해 표준화 기구에 의해 지정 포트가 정의되어 있음(https : 443/tcp)

o SSL을 사용하기 위해서는 우리가 흔히 사용하는 URL 표기 방식인 "http://*" 대신에 "https://*"을 사용해야 함

o SSL은 Record Layer와 Handshake Layer로 구분

- Record layer
  · 메시지 캡슐화
- Handshake Layer
  · 응답확인 방식을 통한 서버와 고객 간의 인증
  · 암호화 알고리즘 선택
  · 암호화 키 계산
- SSL의 문제점
  · 미국의 정책에 의해 국외 판매 제품 512 bit RSA Public Key와 40 bit RC2 Single key 제한
  · 외산 알고리즘 사용시 보안 문제 우려



(그림 3-6) SSL의 Handshake 프로토콜

(4) 전자상거래 지불 프로토콜

(가) SET 프로토콜

VISA와 Master Card사가 신용카드를 기반으로 한 인터넷 상의 전자 결제를 안전하게 이룰 수 있도록 마련한 전자 결제과정 표준안

o 지불 프로토콜 기반의 상거래 절차
o SET 프로토콜의 목적
  - 정보의 기밀성 확보
  - 지불 정보의 무결성 확보
  - 상인과 고객의 상호 확인
o SET 프로토콜의 주요 기능
o SET 프로토콜의 특징
  - 트랜잭션 정보의 비밀성 보장
  - 데이터의 무결성
  - 카드소지자 및 상점 인증
o SET 프로토콜의 장점
  - 전자 거래의 사기를 방지
  - 기존의 신용카드 기반을 그대로 활용
  - SSL의 단점(상인에게 지불정보 노출)을 해결

o SET 프로토콜의 단점

- 암호 프로토콜이 너무 복잡함
- RSA 동작은 프로토콜의 속도를 크게 저하시킴
- 카드소지자에게 전자지갑 소프트웨어를 요구함
- 상점에 소프트웨어를 요구함
- 지불게이트웨이에 거래를 전자적으로 처리하기 위한 별도의 하드웨어와 소프트웨어를 요구함

o SET 프로토콜 인증

-SET(Secure Electronic Transaction)은 전자상거래 시 안전한 지불을 위한 내용을 담고있다

・고객과 Merchant간에 서로의 신분을 확인할 수 있는 인증에 관한 내용

・인터넷 상에서 메세지를 안전하게 주고 받을 수 있는 암호화 기법에 관한 내용

・지불절차에 관한 내용

-이중 서명(Dual Signature)

・구매요구(Purchase Request) 거래에서

・상인은 주문정보만을 알아야 하고

・매입사(Payment Gateway)는 지불정보만을 알아야 함.

・고객이 결재 정보를 상인에게 보낼 때

・주문정보는 상인의 공개키를 이용 암호화

・지불정보는 매입사(Payment Gateway)의 공개키를 이용 암호화

・이중서명이란 주문정보의 메시지 다이제스트와 지불정보의 메시지 다이제스트를 합하여 다시 이것의 메시지 다이제스트를 구한 후 고객의 서명용 개인키로 암호화한 것을 말 함

・주문정보와 지불정보 각각에는 이중서명과 함께 상대편 정보의 메시지 다이제스트가 포함되어 있다. 따라서, 이 정보를 받은 상인 또는 매입사는 자신이 받은 정보의 메시지 다이제스트를 구한 것과 상대편 정보의 메시지 다이제스트를 합하여 메시지 다이제스트를 다시 구한 후 이중서명을 고객의 서명용 공개키로 푼 것을 비교함으로써 서명을 확인

(5) 전자 지불 시스템 기술 요건

o 거래 상대방의 신원 확인
o 전송 내용의 비밀 유지
o 전자 문서의 위· 변조 및 부인 방지
o 거래 정보에 대한 접근 통제

(6) 전자지불 시스템 위협 요소

o 위조
o 국가통화관리
o 이중사용
o 위장

(7) 안전한 전자지불 서비스를 위한 보안 메커니즘

o 불추적성
o 분할성
o 익명성 제어

2.2.2 전자 입찰 프로토콜

전자 상거래 방식을 통한 공개 구매 및 전자 입찰방식에 대한 개념을 이해하고 입찰부정을 방지하고 공정 경쟁이 가능하도록 하는 전자 입찰 시스템 및 프로토콜의 특징에 대한 이해 정도를 평가하도록 한다. [1,2급 공통출제]

(1) 전자입찰

o 핵심가이드
- 전자 입찰 특징 이해
- 전자 입찰 시스템의 구성 요소 이해
- 전자 입찰 수행시의 문제점 이해
- 전자 입찰 시 요구 사항 이해
- 전자 입찰 도구 등

o 전자 상거래 방식을 통한 공개 구매시 다양한 거래선을 확보할 수 있고 구매원

가가 절감 된다.

o 전자입찰방식은 저렴한 가격으로 입찰에 대한 정보를 웹사이트에 게재하게 되므로 입찰부정의 소지가 없어지고 공정한 경쟁 입찰이 가능해진다.

o 전자 입찰 시스템의 구성 요소
  - 입찰자
  - 입찰 공고자
  - 전자 입찰 시스템

o 전자 입찰 수행시의 문제점
  - 네트워크 상의 메시지 유출
  - 입찰자와 서버 사이의 공모
  - 입찰자간의 공무
  - 입찰자와 입찰 공무자간의 공모
  - 서버의 독단

o 전자 입찰시 요구 사항
  - 독립성
  - 비밀성
  - 무결성
  - 공평성
  - 안전성

o 전자 입찰 도구
  - 자바
  - 디지털서명
  - XML 이용

(2) 전자 입찰 단계

o 핵심가이드
  - 입찰단계의 활동 이해
  - 입찰 오픈 및 입찰자 결정 단계 활동 이해

o 입찰 단계
  - 입찰 내용 제출 : 비밀성
  - 입찰 기간 마감 : 여러 개의 입찰 서버가 있을 경우 동시 마감

o 입찰 오픈 및 입찰자 결정단계

- 입찰 내용 오픈
- 입찰 내용 검증 : 입찰의 정당성 검증
- 입찰자 결정 : 공개적 결정

(3). 전자 입찰 프로세서

o 핵심가이드

- 결재 보안 프로토콜 등
- 전자 메일, 전자 게시판등
- 고객 대응 종류 이해

o SET : 결재 보안 프로토콜

o SCM : EDI, 전자메일, 전자게시판

o QR (Quick Response)

o ECR (Customer Response)

- QR : 섬유산업 즉시 대응
- ECR : 식품/잡화 효율적 소비자 대응
- EHCR : 의약품 효율적 소비자 대응
- EFR : 식품분야 효율적 소비자 대응
- BPR : 개별적인 프로세스의 재구축을 통해 획기적인 성과향상을 목표로 한다. 구성원의 참여가 부분적이다. 비 제조분야에 적합하다.
- TQM : 조직전반의 프로세스의 지속적인 개선을 통해 점진적인 성과향상을 목표로 한다. 구성원의 참여가 전사적이다. 제조분야에 적합하다.
- Downsizing : 재무구조(자산 및 자본구조)의 변화를 수반하지 않는 조직구조의 변화, 조직의 효율성을 목표로 한다.
- Restructuring : 재무구조의 변화를 수반하는 조직구조의 변화. 조직 및 전략의 효율성을 목표로 한다.

2.2.3 전자 투표 프로토콜

전자투표의 필요성이 증대되면서 안전하고 신뢰성을 기반으로 하며 개인정보보호를 목적으로 하는 전자 투표 프토로콜의 특징을 이해하며, 투명서 있는 운영을 위한 다양한 요구사항에 대해 이해하고 평가한다. [1,2급 공통출제]

o 핵심가이드

- 전자 투표 시스템의 이해
- 전자 투표 시스템 요구사항 이해
- 전자 투표 시스템 구성을 위한 필요 암호 기법 이해

o 모든 투표 과정이 인터넷을 통해 이뤄지는 투표시스템
o 철저한 개인인증 절차와 완벽한 보안시스템의 보호 아래 투표하게 되므로 안전하다.
o 투표가 끝남과 동시에 결과가 집계되기 때문에 기존 투표절차에 비해 빠르고 정확하다.

(1) 전자 투표 시스템 요구사항

o 완전성 : 모든 투표가 정확하게 집계되어야 한다
o 익명성 : 투표결과로부터 투표자를 구별할 수 없어야 한다
o 건정성 : 부정한 투표자에 의해 선거가 방해되는 일이 없어야 한다
o 이중투표방지 : 정단한 투표자가 두 번 이상 투표할수 없다
o 정당성 : 투표에 영향을 미치는 것이 없어야 한다
o 적임성(투표자격제한 선거권) : 투표권한을 가진 자만이 투표 할 수 있다
o 검증 가능 : 선거 결과를 변경 할 수 없도록 누구라도 투표 결과를 확인하여 검증해 볼 수 있어야 한다

(2) 전자 투표 시스템에 필요한 암호 기법

o 공개키/개인키를 이용한 암호화/복호화 함수
o 전자서명
o 은닉암호

2.3 무선 플랫폼에서의 전자상거래 보안

무선 플랫폼에서의 전자상거래 보안을 위한 기반 기술을 이해하고 각종 무선 전자상거래 서비스를 제공하기 위한 특징에 대해 평가하도록 한다. 그리고 무선 컨텐츠 지불 서비스 모델을 구성하고 있는 보안 프로토콜과 지불 프로토콜에 대한 이해를 평가한다.

o 무선 전자상거래(m-commerce)는 이동통신 네트워크 기술과 무선단말기를 기반으로 하여 언제 어디서나 필요한 시점에 행할 수 있는 상거래를 의미한다. 즉, 모바일 폰(hand-held phone), PDA, 노트북 등 무선단말기를 이용하여 B2B, B2C를 비롯하여 컨텐츠, 정보제공, 오락, 게임 등을 포함하는 모든 유료화된 상거래를 의미한다.

(1) 무선 전자 상거래 서비스 주요 특징

o 핵심가이드

- 무선 전자 상거래 주요 특징 이해
- 무선 전자 상거래 서비스 이해
- 모방일 무선 단말기의 제약 사항 이해
- 무선 네트워크의 제한 사항 이해

o 무선 전자상거래의 주요 특징

- 편재성
- 도달성
- 보안
- 편리성
- 위치성
- 즉시 접속
- 개인화

o 무선 전자상거래 서비스

- 커뮤니케이션 서비스(Communication Service)
- 정보 서비스(Information Service)
- 엔터테인먼트 서비스(Entertainment Service)
- 거래서비스(Transaction Service)

o 모바일 무선 단말기의 제약사항

- CPU 및 OS 성능의 제한
- 메모리의 제한
- 이동성을 위한 소비 전력의 제한
- 제한된 디스플레이창
- 입력 방식의 제한

o 무선 네트워크의 제한 사항

- 작은 대역폭 (이동전화 전송 속도 : 9.8Kbps ~ 14.4Kbps)

- 더 큰 신호 도달 시간
- 낮은 접속 안정성
- 높은 비용

(2) 무선 컨텐츠 지불 서비스 모델

o 핵심가이드

- 무선 컨텐츠 지불 서비스 모델 종류 이해
- 전자지불 시스템 프로토콜 특징 이해

o B2B 전자지불 서비스 모델

o 신용카드 기반 전자지불 시스템

- 보안 프로토콜 : END-TO-END간에 발생하는 Transaction의 안전성 보장
  - S-HTTP(Secure-HTTP)
  - SSL(Secure Socket Layer)
  - TLS(Transport Layer Security)
- 지불 프로토콜 : 전자상거래 행위에 관련된 모든 구성원들 간의 트랜잭션을 정의하고 안전성 보장을 위한 별도의 프로토콜을 설계함으로써 전자상거래 지불시 안전성 보장
  - SET(Secure Electronic Transaction)
  - InstaBuy(CyberCash)

o 전자화폐 기반 전자지불 시스템

- 네트워크형 프로토콜 : 인터넷과 같은 네트워크 환경에서 사용자의 PC나 서버의 계좌 등에 자신의 전자화폐를 저장하고, 사용하는 형태의 프로토콜
  - Millicent(DEC)
  - ·NetBill(CMU)
  - Payword(MIT)
  - E-cash(Digicash)
  - 기타(InterCoin, Subscrip, PayMe, MiniPay, iKP 등)
  - 국내(Ecoin, iCash, EasyCash)
- 가치저장형 프로토콜 : 스마트카드 내에 전자화폐를 저장하고 사용하며, 인터넷과 같은 네트워크 환경보다는 실생활의 화폐를 대용키 위한 목적으로 구성된 프로토콜들

· ·Mondex(MasterCard)
· VisaCash(Visa International)
· Proton(Banksys)
· ChipKnip(Interpay B.V.)
· 국내(K-Cash 등)

o 전자수표 기반 전자지불 시스템
- 전자수표 프로토콜 : 실세계의 수표와 유사한 형태로 전자서명과 같은 암호 기술을 사용함으로써 배서 등의 효과 제공
· Echeck(FSTC)
· NetCheque(USC)
· PayNow(CyberCash) 등

2.3.1 무선플랫폼에서의 전자상거래 보안[1급]

o 핵심가이드
- WML 스크립트 암호화 보안 프로토콜 이해
- 무선 공개키 기반 구조 이해
- 무선 공개키 기반 구조의 구성요소 이해

o WMLScript Crypto 라이브러리 보안 프로토콜
- WMLScript를 이용한 서명
- WMLScript를 이용한 암/복호화 솔루션 (vodafone, telstar, certicom)
- WMLScript를 이용한 secure session

o WPKI(Wireless Public Key Infrastructure)
- WAP에서 서버와 클라이언트 간의 인증을 위한 무선 환경에 적합한 인증서를 발급, 운영, 관리하는 무선망의 공개키 기반 구조를 말한다.
- 현재 유선 인터넷 상에서의 PKI는 빠르게 이루어져 있어 증권, 금융, 쇼핑 등의 전자 상거래가 이루어질 수 있는 핵심 분야에 대한 공인인증기관이 지정되어 있고, 이를 통한 일부 사용서비스가 이루어지고 있다.

o WPKI 구성
- CA 서버 시스템 : 인증서 발급, 관리
- RA 서버 시스템 : 인증서 발급, 관리 요청 중계
- Client 시스템 : 인증서 발급, 관리 요청

- Direct 서버 시스템 : CA가 발행한 인증서 정보를 저장, 관리

2.4 전자상거래 응용 보안

e-Business 상에서 인터넷 표준 브라우저를 통해 장소에 구애없이 전자상거래를 위한 응용 분야 기술들에 대한 이해를 평가한다. 그리고 ebXML 보안에 대한 이해를 평가한다.

2.4.1 e-business를 위한 ebXML 보안 [1급]

- o 전자상거래 기반 기술을 토대로 특정한 응용분야 기술들이 개발되고 있는데 그 중에 하나가 ebXML(electronic business XML)이다. 이는 인터넷 표준 브라우저만으로 장소에 구애 없이 어디서나 전자상거래를 할 수 있으며 저렴한 구현 비용, 개방된 네트워크로 전자거래 교환을 위한 국제 표준을 제공한다.
- o 기존의 EDI(Electronic Data Interchange)와는 달리 XML에 기반하고 있어 각각의 시스템을 가진 다양한 업종의 회사들간에 무수한 형태의 계약을 전자상거래로 처리할 수 있다.

가. 전자문서

전자문서의 등장으로 상거래 및 정보교환의 상당부분이 전자적으로 대체되었다. 이러한 업무 성격 변화가 가져오는 영향에 대한 이해를 평가 한다

o 핵심가이드

- 전자 문서 이해
- 전자 거래 문서의 유형 이해

(1) 전자문서 개요

오늘날 컴퓨터 및 통신 기술의 급속한 발달은 그것이 상거래이든 단순한 정보교환이든 상관없이 기존의 종이문서에 의한 방식을 상당 부분 전자적 방법으로 대체하고 있다. 이와 같은 전자문서의 등장은 단순히 전통적인 종이문서의 형식만을 바

꾸는데 그치지 않고 문서 및 업무의 성격에도 중대한 변화를 초래하고 있다.

(2) 전자 거래 문서의 유형

ebXML, 웹서비스,로제타넷, SWIFT, 볼레로 등의 XML 기술을 적용한 문서가 개발되고 있으며, 특히 ebXML은 국제 전자거래문서 표준으로서 폭넓게 채택되고 있다.

o EDI(Electronic Data Interchange) 문서
  - 기업간 거래에 관한 데이터와 문서를 표준화하여 컴퓨터 통신망으로 거래 당사자가 직접 송·수신하는 정보전달 시스템이다. 주문서·납품서·청구서 등 무역에 필요한 각종 서류를 표준화된 상거래서식 또는 공공서식을 통해 서로 합의된 전자신호로 바꾸어 컴퓨터 통신망을 이용하여 거래처에 전송한다.
  - 전자문서교환에서 사용하는 국제적인 통신표준은 현재 국제연합이 중심이 되어 만든 UN/EDIFACT의 표준을 따르고 있다.

o XML/EDI 문서
  - XML/EDI는 EDI를 통하여 교환된 데이터를 XML기반 타 업무 프로세스에 바로 적용될 수 있는 개방적 구조를 가지기 때문에 업무 효율성의 제고 등 실질적인 EDI 도입의 효과를 기대할 수 있으며, 값싼 구축/운영비용과 인터넷을 바로 이용할 수 있다는 장점이 있다.

o XML(Extensible Markup Language) 문서
  - XML은 데이터의 저장 및 교환을 위한 대표적 문서교환 표준인 SGML(Standard Generalized Markup Language)과 HTML(Hyper Text Markup Language)의 장점을 모두 가질 수 있도록 1996년 W3C(World Wid Web Consortium)에서 제안하였으며 웹상에서 구조화된 문서를 전송 가능하도록 설계된 정보 교환을 위한 웹 표준이며, 최근 전자거래 및 각종 업무에서 표준으로 폭넓게 채택되어 사용되고 있다.

나. e-Business를 위한 ebXML

기업간 전자거래가 과거 EDI 환경에서 인터넷 환경으로 변화되어가면서 시스템

간 통합이나 전자거래표준들이 나오고 있다. 이러한 e-Business 환경에 대해 이해하고 B2B 표준에 대한 요구사항 및 관련 기술에 대한 이해를 평가한다.

o 핵심가이드

- ebXML의 필요성 이해
- B2B 표준을 위한 요구 사항 이해
- ebXML(Electronic Business Extensible Markup Language)의 정의

o 기업간 전자거래는 업무 프로세스로부터 시작하여, 전자문서(전자문서를 구성하는 데이터 항목 및 코드), 협력을 위한 프로파일과 전송 프로토콜, 보안 체계 등 거의 전 부문에서 서로간에 합의를 전제로 해야 가능하다.

o 개별 몇 개 업체간의 지역적인 합의에 의한 전자거래는 비교적 쉽게 가능 할 수도 있지만 인터넷이 기본이 되는 글로벌한 거래 파트너들을 대상으로 전자거래를 하기 위해서는 기업간 전자거래에 있어 국제 표준의 역할이 필수적이라 할 수 있다.

o 기업간 전자거래는 과거 EDI를 기반으로 하고 있으나, 최근 인터넷 환경에 서 XML표준을 적용하여 원활한 거래를 하기 위한 표준의 중요성이 계속 강조되고 있다. 현재, 시스템간 통합이나 일부 산업용 전자거래 표준들이 나와 있지만, 모든 산업에 적용가능하며 기업간 전자거래를 목적으로 하는 표준 체계는 ebXML이 유일하다 할 수 있다.

o 우리나라에서는 이미 업계에서 ebXML표준을 적용하여 저렴하고 안정적인 기업간 거래를 검증하였고 여러 산업부문에서 대기업은 물론 중소기업까지 확산되고 있다.

(1) B2B 표준을 위한 요구 사항

e비즈니스(B2B)거래 표준적용을 위하여 다음과 같은 업무 요구사항을 기술하고, 이를 충족하기 위해서 ebXML을 도입하도록 적극 권고한다.

o 전자문서
o 메시징
o 정보 등록 저장
o 협약

o 비즈니스 프로세스
o 보안
o 카탈로그
o 기업내부시스템 연동

(2) ebXML(Electronic Business Extensible Markup Language) 정의

UN/CEFACT와 OASIS가 주도하여 기업의 규모나 지역적 위치에 관계없이 인터넷을 통해 거래 할 수 있도록 하는 규약들의 모음. (거래 메시지의 교환, 거래 메시지의수립, 공통출제의 조건에 의한 데이터 통신, 비즈니스 프로세스 정의 및 등록)

o 2001년 6월 국내 전자상거래 표준 프레임워크로 채택
o 2003년 5월 UN 전자상거래 국제표준으로 승인
o 2004년 3월 ISO ebXML의 4개 기술 규격에 대한 표준으로 승인

다. ebXML 구성

e-Business 환경 상에서 기업간 전자거래를 위한 표준 체계인 ebXML의 특징 및 구성요소, 특징적 요소 그리고 사용 효과에 대한 이해를 평가 한다.

o 핵심가이드
- ebXML 구성 요소 이해
- ebXML 특징 이해
- ebXML 특징적 요소 이해
- ebXML 사용 효과 이해

(1) ebXML 구성 요소

ebXML 이 Biztalk 등 XML 기반의 다른 프레임워크와 구별되는 부분은 거래파트너 사이에서 주고받는 XML메시지만을 규정하는 것이 아니라 비즈니스 프로세스 모델, 핵심 컴포넌트 집합은 물론 분산된 레지스트리의 구축까지 규정한다.

o 비즈니스 프로세스 (Business Process) : 다양한 비즈니스 거래절차에 대한 내

용을 표준화된 방법으로 모델링해 시스템이 자동으로 인식, 처리할 수 있도록 하는 표현방법에 대한 정의 규정한다.

- o 핵심 컴포넌트 (Core Components) : 비즈니스에서 교환되는 전자문서(메시지)를 이루는 항목을 미리 잘 정의해 재사용 가능하도록 표준화 작업을 한다.
- o 등록저장소 (Registry/Repository) : 저장소는 거래상대자들에 의해 제출된 정보를 저장하는 안전한 저장소이며, 등록기는 이들 정보의 메타데이터를 등록시켜 놓은 등록소이다. 등록저장소분야는 전체 ebXML의 가장 중요한 부분이라고 할 수 있다.
- o 거래당사자 (Trading Partners) : 비즈니스 거래 당사자에 대한 각종 정보 및 협업을 위한 프로파일을 통일된 규칙으로 표현하며, 이러한 내용을 CPP(협업 규약 프로파일 : 거래당사자의 정보), CPA(협업 규약 약정서 : 거래상대자들 간의 협약)로 표현한다.
- o 전송, 교환 및 패키징 (Transport/Routing and Packaging) : ebXML메시지서비스를 제공하여 메시지를 상호운용성과 보안을 유지하면서 어떻게 전달할 것인가에 대한 표준을 정립한다.

(2) ebXML의 특징

- o 개방성(명세개발 작업에 있어 누구나 아무런 비용부담없이 참여 가능)과 상호연동성(ebXML 명세에 따라
- o 누구나 특정 솔루션이나 플랫폼에 의존없이 전자상거래가 가능)
- o 개방적인 XML을 범세계적으로 제공하여 국제적이고 단일한 e-Marketplace 제공
- o 상호연동성이 가능한 매우 유연한 솔루션으로 e-비즈니스 범위 확장
- o 현재 새로운 XML의 다양한 시도들을 흡수하고 이를 촉진

(3) ebXML의 특징적 요소

- o 글로벌한 광고 가능
- o 분산된 형태의 아키텍처
- o 리치 쿼리 기능이 가능
- o 믿을 수 있는 SOAP를 이용
- o 보다 특화된 그룹에 신경을 씀.

o 벤더 독립적
o 비즈니스 프로세스에도 신경 쓰는 표준
o ACID 개념이 전혀 없음
o spec뿐이기에 벤더 보장, commitment가 없음

(4) ebXML 사용 효과

o 재활용성
- XML 문서는 그 구성이 각각의 엘리먼트('<,>'(태그)로 둘러싸인 항목), 즉 컴포넌트로 구성이 된다.
- 컴포넌트들 중 모든 문서에서 공통출제적으로 사용되는 컴포넌트들을 골라내어 핵심컴포넌트를 만들고, XML문서를 만들 때 핵심컴포넌트를 사용하여 XML 문서를 만든다.
- XML 문서를 만들 때마다, 컴포넌트를 만드느라 고민할 필요 없이, 만들어져 있는 핵심컴포넌트에서 필요한 항목을 뽑아 쓰기만 하면 되므로, 문서개발에 중복되는 시간과 비용을 대폭 절약할 수 있으며, 문서의 구성도 표준화시킬 수 있다.

o 비즈니스프로세스 활용
- 기존의 표준들은 대부분 단순히 문서만을 표준화하여 사용하는 것과는 달리, ebXML 에서는 재활용의 수준을 문서수준뿐만 아니라, 시나리오 수준까지 확대해 사용한다.
- 비즈니스 전 과정을 모델링하여 시나리오를 작성하고, 이 시나리오에 따라서 B2B거래를 자동화해 실행한다.

라. e-Business를 위한 ebXML 보안

현재의 e-Business환경에 대해 이해하고 인터넷 서비스를 통한 글로벌 환경의 서비스지원 위한 보안요구사항에 대해 평가한다. 그리고 서비스를 제공할 수 있는 ebXML 표준 체계의 보안 요구사항과 관련 XML 기술에 대해 평가 한다.

o 핵심가이드

- 현재의 e-Business 특징 이해
- ebXML & Web Services 환경 이해
- ebXML & Web Services 보안 요구 사항 이해
- 주요 XML 기반 보안 기술

(1) 현재의 e-Business

o 자국내의 영역에서 벗어나 글로벌한 환경으로의 확대
o 국가간의 거래시 상이한 비즈니스 프로세스로 인한 비용증가

(2) ebXML & Web Services 환경

o 글로벌한 단일 비즈니스 환경 구축을 목표
o 국내외 적으로 세부 컴포넌트 구현 기술 연구 활발

(3) ebXML & Web Services 보안 요구

o 무결성, 기밀성, 부인방지, 인증 등의 거래 신뢰성 보장
o 보안 요구사항은 XML 기반 보안 기술 적용
  - XML 전자서명, XML Encryption, XKMS, SAML, XACML

(4) ebXML에서의 보안 요구사항

o W3C/OASIS 등에서 표준화가 진행 또는 완료된 기술적용

(5) 주요 XML 기반 보안 기술

o XML 전자서명(XML Signature) : 영구 무결성 및 부인 방지
o XML Encryption : 영구 기밀성
o XKMS(XML Key Management Specification) : PKI 서비스 프록시
o SAML(Security Assertion Markup Language) : 인증,속성, 승인 Assertion
o XACML(eXtensible Access Control Markup Language) : XML기반 접근 제어

3. 기타 어플리케이션 보안

3.1 응용프로그램 보안개발 방법

일반적으로 많은 OS 응용프로그램에서는 버그(bug)라 불리우는 보안 취약점이 포함되어 있고 이 취약점에 대한 보안패치(patch)를 통해 해당 취약점을 제거하도록 보안권고하고 있다. 이런 보안대책을 따르지 않을 경우 매우 심각한 피해를 입기도 한다. 따라서 응용프로그램을 개발함에 있어서 이와 같은 보안 취약점을 남기지 않고 개발하는 보안프로그래밍이 매우 중요하며, 특히 최근 많이 등장하는 버퍼오버플로우(Buffer Overflow)와 포맷스트링(Format String) 버그를 방지할 수 있는 프로그래밍 기술을 익히는 것은 보안프로그래밍의 기본이라 할 수 있다. [1급]

o 핵심가이드

- 취약점 및 버그방지 개발방법의 종류
- 메모리 구조의 이해
- 버퍼오버플로우와 포맷스트링 취약점의 개념 이해
- 각 취약점에 대한 주요 보안대책의 이해
- 프로그래머 관점의 개발시 사용 제한이 필요한 함수의 종류
- 프로그램 작성 시 사용 제한 및 유의해야 할 포맷함수의 종류
- 포맷스트링 취약점 점검툴의 종류

3.1.1 취약점 및 버그방지 개발방법[1급]

현재 십여 가지의 운영 시스템과 수십만 개의 다양한 응용 프로그램들이 있다. 그리고 역대로 그러한 운영 시스템과 프로그램들은 취약점을 항상 가지고 있었다. 하지만 많은 취약점들은 기존의 공격 방식들을 변형시켜서 공격할 수가 있다.

(1) 주요 보안대책

o SUID/EUID 보안프로그래밍

- UID와 GID의 제한 권장
- exec를 호출하기 전에 유효(effective) UID와 GID를 재설정
- exec를 호출하기 전에 모든 파일기술자(descripter)를 닫음

o 새로운 프로세스의 생성보안
- system(), popen() 함수의 사용 금지
- 모든 파일 기술자를 닫았는지 확인
- 프로그램을 실행할 때 전체 경로 이름을 사용하는지 확인
- 자식프로세스에 전달된 환경변수를 확인

o 안전한 임시 파일 사용보안
- 알려진 임시 디렉토리(예: /tmp) 안에 임시 파일을 생성 금지
- 임시파일을 생성하는 인터페이스를 제공하는 시스템을 사용
- 예측 가능한 임시파일 이름을 생성하지 말고 랜덤하게 생성

o 버퍼오버플로우 방지 프로그래밍
- 안전한 함수 사용

(2) 버퍼오버플로우의 이해

스택 버퍼 오버플로우 공격은 오늘날 굉장히 일반화되었고, 공격자들에게 권한을 얻을 수 있도록 도와주며, 취약한 시스템을 마음대로 조종할 수 있게끔 만들어 준다. 이 공격 방법은 오래 전부터 알려져 왔었지만 Aleph One이 Phrack에 독창적인 글을 쓴 이후에 이 공격 방법은 더욱 유명해 졌다.

o 버퍼오버플로우 개념

- 글자 그대로 버퍼(Buffer)가 넘친다는 의미
- 지정된 메모리의 양보다 더 많은 양의 데이터를 쓰려고 할 때 발생
- 버퍼 오버플로우 공격을 통하여 스택(Stack)에서 리턴 어드레스(Return Address)를 수정하면 프로세스의 흐름을 조정할 수 있다.

컴퓨터 프로그램을 실행시키는 과정과 컴퓨터가 프로그램을 실행시킬 때 메모리에서 일어나는 일 등을 자세하게 이해하고 있어야 공격방법을 이해할 수 있는 어려운 기술 중의 하나이다.

o 버퍼오버플로우의 종류

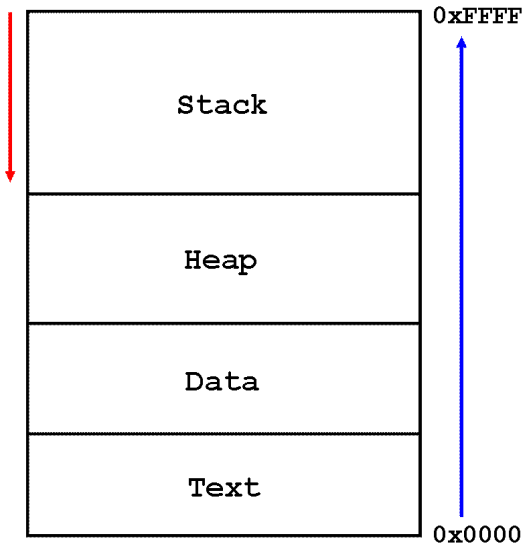
- 스택(Stack) 오버플로우 공격 : 스택영역에 할당된 버퍼 크기를 초과하는 데이터(실행가능코드)를 기록하고 저장된 복귀주소를 변경함으로써 임의의 코드를 실행

- 힙(Heap) 오버플로우 공격 : 힙영역에 할당된 버퍼 크기를 초과하는 데이터(실행가능코드)를 기록하고 저장된 데이터 및 함수의 주소를 변경함으로써 임의의 코드를 실행

o 프로세스와 메모리 구조

Process는 실행 되어 지고 있는 프로그램이다. 실행 되어 지고 있다는 말은 메모리에 그 프로세스가 이용하고 있는 구역이 존재한다는 뜻이기도 하다. Process가 사용하는 메모리 영역(Process Address Space)은 서로 다른 Process의 영향을 받지 않고 주지 않기 위해서 자신이 혼자 4Giga Byte의 영역을 혼자 사용하는 듯한 착각에 빠지도록 만드는 Virtual Address라는 방식을 사용한다. Logical address를 Physical address로 변환하여 사용하는 형태이다.

이 영역은 Binary를 실행하는 Language와 Compiler에 따라서 다르게 사용되는데 일반적으로 사용하는 C Compiler가 생성하는 Binary는 다음과 같은 형태로 Process의 Memory구조를 형성한다.



(그림 3-7) 메모리 구조

Text 영역

- 프로그램의 본체(명령 코드들의 집합)와 Read-Only Data들을 담고 있다.
- 따라서, 프로그램이 Text로 되어있는 메모리 영역을 침범하여 기록을 하려 한다면 Bus Error나 Segmentation Fault가 일어나서 프로그램이 종료된다.

Data 영역

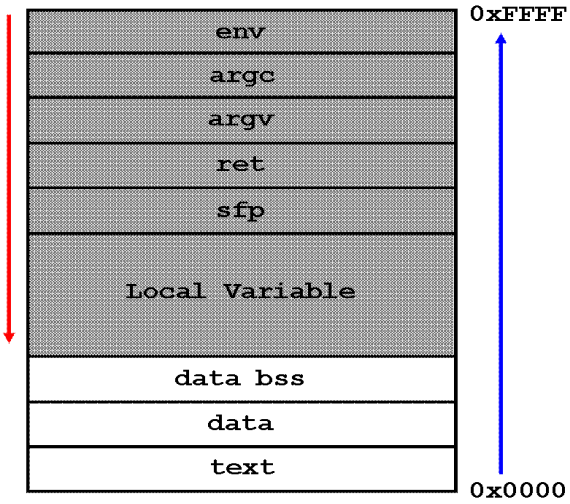
- C 언어에서 전역변수, 정적변수 등으로 선언되는 변수들을 기록하는 목적으로 사용되어진다.
- 이곳은 읽고 쓰기가 가능한 영역이다.

Heap 영역

- 정돈되어있지 않은 공간
- 어떻게 사용되어 질지도 모르는 상태로 가상적으로 존재하는(Allocation되지 않은) 공간이다.
- Uninitialized Data Region으로도 불리고 있는 Heap은 프로그램 수행 중 malloc 등의 System Call로 할당되어 사용되어지다가 free System Call로 Free되는 등 자유자재로 사용가능 영역이다.

Stack 영역

- C가 Procedure Calling(Function Calling)이 가능한 언어이기 때문에 만들어진 영역으로 Program의 수행 중 Function Call이 있을 경우 Stack에 새로운 Function에서 사용되어 질 지역변수, 파라미터 변수, 함수가 끝났을 경우 리턴할 명령 포인터 값을 저장하는 Return Address등을 Push하게 된다. 이 후 함수가 끝났을 경우 위의 값들을 POP하고 Return하게 되는 것이다.



(그림 3-8) 메모리에 적재된 모습

0x0000 부분이 메모리의 낮은 부분이고 0xFFFF 부분이 메모리의 높은 부분이다. 그리고 음영처리 된 부분이 스택을 나타낸다. 데이터가 메모리에 적재될 때는 메모리의 낮은 주소에서 높은 주소로 적재 되고 스택은 그와 반대이다.

env : 프로그램이 실행될 때 사용하는 환경변수의 값
argc : 프로그램의 인수의 개수
argv : 프로그램의 인수 값
ret : 프로그램이 끝나고 돌아가야 할 주소
sfp : Stack Frame Pointer로 현재 스택의 위치 값
Local Variable : 지역변수
data bss : 초기화 안된 전역 변수
data : 초기화 된 전역 변수
text : 실행 명령

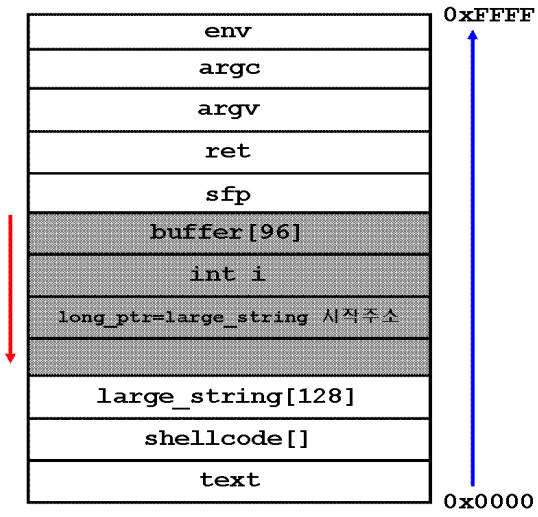
o 버퍼오버플로우 코드의 예

```
char shellcode[] =
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

```
char large_string[128];

void main() {
  char buffer[96];
  int i;
  long *long_ptr = (long *) large_string;

  for (i = 0; i < 32; i++)
    *(long_ptr + i) = (int) buffer;

  for (i = 0; i < strlen(shellcode); i++)
    large_string[i] = shellcode[i];

  strcpy(buffer,large_string);
}
```

먼저 shellcode[] 부분은 data 영역에 들어가게 된다. 전역변수이므로 메모리의 낮은 주소에서부터 적재되는 것이다. 그리고 large_string[128] 변수는 아직 초기화가 되지 않은 전역변수이므로 data bss 부분에 적재되게 된다. 이후 main() 함수가 시작하면서 변수들이 스택에 적재되게 된다.

변수들이 메모리에 적재된 모습은 다음과 같다.



(그림 3-9) 메모리에 적재된 모습

```
for (i = 0; i < 32; i++)
    *(long_ptr + i) = (int) buffer;
```

long_ptr주소에 i를 더한 주소의 내용, 즉 buffer의 시작주소를 int형태로 large_string 배열에 저장한다. 참고로 주소는 4byte이다. 따라서 i 가 1이라면 long_ptr의 주소에서 4byte만큼 떨어진 값인 lagre_string의 주소에 buffer주소를 넣는다.

```
for (i = 0; i < strlen(shellcode); i++)
    large_string[i] = shellcode[i];
strcpy(buffer,large_string);
```

large_string의 내용을 buffer에 저장한다. shellcode가 있는 46byte를 제외한 나머지는 모두 buffer배열의 시작주소가 있다고 했다. 결과적으로 ret부분에 buffer의 시작주소가 들어가게 된다.

프로그램이 종료되면 ret값을 참조하는데 buffer의 시작주소를 참조하여 실행하게 된다. buffer 배경의 처음 위치부터는 shellcode가 들어 있어 실행하게 되는 것이다.

버퍼 오버플로우의 핵심은 shellcode이다. 이 shellcode는 운영체제마다 다르며 shellcode를 만드는 것 또한 쉽지는 않다.

(3) 버퍼오버플로우 보안대책

o 운영체제 커널 패치

사용자 스택 영역에 데이터 기록 금지하게 한다. 그리고 함수로부터 복귀할 때 스택의 무결성(integrity) 검사하게 하며 된다. 그리고 코드의 실행을 금지시키는 방법이 있다.

리눅스에서는 아래의 사이트에서 패치를 받아서 설치하면 된다. Openwall 프로젝트 중 하나로 사용자 스택 공간에서 실행을 하지 못하게 하는 패치이다.
http://www.false.com/security/linux-stack

Solaris 2.7 이상의 버전에서는 /etc/system 파일에 다음 두 라인을 추가하여 버퍼 오버플로우를 막을 수가 있다.

set noexec_user_stack = 1
set noexec_user_stack_log =1

o 경계 검사를 하는 컴파일러 및 링크 사용

GNU GCC 2.7.2.3 버전을 패치한 StackGuard를 사용한다. 이 컴파일러는 복귀 주소 다음에 “canary word를 위치시킴으로써 “canary” word가 변경되면 스택 오버플로우 공격의 지도 감지 및 보고(syslog)한다.

Random canary는 실행할 때마다 canary value를 변경시켜, 공격자가 예측하지 못하도록 하게 하는 것이고 Null canary(0x00000000)는 공격자가 버퍼에 널 문자(0x00)를 넣을 수 없다는 점을 이용하여 공격에 대응하는 방법이다.

Terminator canary(combination of Null, CR, LF, -l)는 NULL 문자로 끝나지 않는 몇몇 문자열 함수의 문자열 끝문자 이용한 방법이다.

o 프로그래머의 관점에서의 보안 대책

버퍼 오버플로우 공격을 막는 가장 중요한 방법이다. 최초 어플리케이션 구현시 Boundary를 검사하는 컴파일러 및 링커 사용하여 버퍼 오버플로우 공격에 취약하지 않게 만든다. 또는 Boundary를 검사하는 함수 사용한다.

사용자제를 권장하는 함수들은 다음과 같다.

strcat(), strcpy(), gets(), scanf(), sscanf(), vscanf(), vsscanf(), sprintf(), vsprintf(), gethostbyname()

그리고 사용을 권장하는 함수들은 다음과 같다.

strncat(), strncpy(), fgets(), fscanf(), vfscanf(), snprintf(), vsnprintf()

마지막으로 버퍼 크기를 검사한 후 원래의 프로그램 수행하도록 구현하는 것이다.

o 보안프로그래밍의 예

다음 두 개의 코드의 실행 결과는 똑같다. 하지만 버퍼 오버플로우 공격이 있을 경우 한 코드는 취약하지만 나머지 하나의 코드는 그렇지 않다.

insecure.c

```
#include <stdio/h>
#include <string.h>
#include <ctype.c>

int main(int argc, char **argv) {
        char buffer[1024];
        if(argc > 1) {
                strcpy(buffer, argv[1]);
        }
        printf("buffer: %s\n", buffer);
}
```

secure.c

```
#include <stdio/h>
#include <string.h>
#include <ctype.c>

int main(int argc, char **argv) {
        char buffer[1024];
        if(argc > 1) {
                if(strlen(argv[1]) >= 1023) {
                        fprintf(stderr, "too long\n");
                        exit(0);
                }
                strcpy(buffer, argv[1]);
        }
        printf("buffer: %s\n", buffer);
}
```

(4) 포맷스트링 취약점과 보안대책

과거의 모든 취약점들은 일종의 버퍼오버플로우였기 때문에 포맷스트링 취약점과 비교하여 이해하는 것이 좋다.

버퍼오버플로우는 1980년대 알려져서 1990년대 그 위험이 인식되었고 현재까지도 아주 많은 익스플로잇(Exploit)이 존재하고 있는 취약점이다. 반면 포맷스트링은 1999년 6월에 알려져서 2000년 6월에 그 위험이 인식되어 온 버퍼오버플로우보다는 가장 최신의 취약점이라 말할 수 있다.

포맷스트링은 버퍼오버플로우에 비해 낮은 기술수준이며 그 취약점도 쉽게 찾을 수 있고 일반적인 프로그래밍 지식 수준에서 버그 발생을 줄이는 노력만으로도 취약점을 완전히 줄일 수 있다. 결론적으로 포맷스트링을 이해하고 프로그래밍 버그를 사전에 줄이는 노력만으로 충분한 보안대책이 마련될 수 있다는 의미이다.

이 출제영역에서는 포맷 함수들의 용법에서 전형적인 취약점과 정확한 용법, 그 파라미터의 몇 가지, 그리고 포맷 스트링 취약점의 일반적인 개념에 대해서 이해하고 취약점을 찾는 방법을 통해 기존 어플리케이션에 대한 포맷스트링을 취약점 점검 능력을 갖추고 있어야 할 것이다. [1급]

o 포맷 함수

포맷 함수는 특별한 종류의 ANSI C 함수로서 독립변수를 가지며, 그것으로부터 하나가 포맷 스트링이다. 그 함수가 포맷 스트링의 값을 구하는 동안 그것은 그 함수에 주어진 특정 파라미터에 접근한다. 그것은 역함수이며, 그 역함수는 사람이 읽을 수 있는 스트링 표현으로서 원시 C 데이터 타입을 나타내기 위해 사용된다. 그것들은 정보를 출력하고, 에러 메시지를 프린터하고, 또는 스트링을 처리하기 위해 거의 어떤 C 프로그램에서도 사용된다.

o 포맷 스트링 취약점

만약 어떤 공격자가 포맷 스트링을 ANSI C 포맷 함수에 부분적으로 또는 전체적으로 제공할 수 있다면 포맷 스트링 취약점이 존재한다. 그렇게 함으로써 포맷 함수의 행동이 변하게 되고, 공격자는 목표 어플리케이션에 대한 통제권을 장악할 수 있다. 아래의 보기에서 스트링 user가 공격자에 의해 제공되는데, 공격자는 예를 들어 명령 라인 파라미터 사용을 통해 전체 ASCIIZ-string을 통제할 수 있다.

잘못된 용법:

```
int func (char *user)
{
printf (user);
}
```

Ok:

```
int func (char *user)
{
printf ("%s", user);
}
```

o 포맷 함수 계열

많은 포맷 함수들이 ANSI C 정의에 정의되어 있다. 더 복잡한 함수들이 기반을 두고 있는 몇 가지 기본적인 포맷 스트링 함수들이 있으며, 그것들 중 몇 가지는 표준은 아니지만 널리 사용될 수 있다.

실제 계열 요소:
- fprintf: FILE 스트림에 프린터
- printf: ‘stdout’ 스트림에 프린터
- sprintf: 어떤 스트링 안에 프린터
- snprintf: 길이 확인과 더불어 스트링 안에 프린터
- vfprintf: va_arg 구조로부터 FILE 스트림에 프린터
- vprintf: va_arg 구조로부터 ‘stdout’에 프린터
- vsprintf: va_arg 구조로부터 스트링에 프린터
- vsnprintf: va_arg 구조로부터 길이 확인과 함께 스트링에 프린터 관련요소:
- setproctitle: argv[] 설정
- syslog: syslog 장치에 출력
- err*, verr*, warn*, vwarn* 등

o 포맷 함수들의 사용

이 취약점이 C 코드 어디에서 일반적인지 이해하기 위해 포맷 함수의 목적을 알아보아야 한다.

기능성
- 간단한 C 데이터타입을 스트링 표시로 전환하기 위해 사용됨
- 그 표시의 포맷을 지정하는 것을 허용
- 그 결과로 나오는 스트링을 처리(stderr, stdout, syslog 등에 출력)

포맷 함수의 작동 방식
- 포맷 스트링은 함수의 행위를 통제한다.
- 프린터되어야 하는 파라미터의 타입을 지정한다.
- 파라미터는 스택에 저장된다.(pushed)
- 직접적으로든(value로) 아니면 간접적으로든(reference로) 저장된다. 함수호출은 포맷 함수가 리턴 될 때 그것이 얼마나 많은 파라미터를 스택에 push하는지 알아야 한다.

o 포맷 스트링의 예

포맷 스트링이란 텍스트와 포맷 파라미터를 포함하는 ASCIIZ 스트링이며 다음과 같은 예를 통해 이해될 수 있다.

예:
printf ("The magic number is: %d₩n", 1911);
프린터 될 텍스트는 “The magic number in:이며, 포맷 파라미터 %d가 따르고, 이것은 출력 시 파라미터(1911)로 대체된다. 그러므로 출력물은 “The magic number is: 1911”이다.

o 포맷 스트링 취약점의 종류

[타입 1] 리눅스의 rpc.statd, IRIX의 telnetd 등에서처럼, 여기서 취약점은 syslog 함수의 두 번째 파라미터에 있다. 이 포맷 스트링은 부분적으로 usersupplied이다.
char tmpbuf[512];
snprintf (tmpbuf, sizeof (tmpbuf), "foo: %s", user);
tmpbuf[sizeof (tmpbuf) - 1] = '₩0';
syslog (LOG_NOTICE, tmpbuf);

[타입 2] wu-ftpd, Qualcomm Popper QPOP 2.53에서처럼 여기서 부분적 또는 완전히 usersupplied 스트링은 포맷 함수에 간접적으로 전달된다.
int Error (char *fmt, ...); ...
int someotherfunc (char *user)
{ ...
Error (user);
... }
...

첫 번째 타입의 취약점들은 자동화된 툴들(예를 들어, pscan 또는 TESOgcc)에 의해 안전하게 탐지될 수 있는 반면, 두 번째 타입의 취약점들은 탐지 툴이 함수 “Error”가 포맷 함수처럼 사용된다는 것을 알고 있을 때만 발견될 수 있다.

o 포맷스트링 취약점의 위협요소

- 프로그램의 파괴

포맷 스트링 취약점을 사용한 간단한 공격은 프로세스가 죽게 만드는 것이다. 이것은 예를 들어 core를 덤퍼하는 데몬을 죽이는데 유용하게 사용될 수 있다. 또는 몇몇 네트워크 공격에서 DNS 스푸핑을 할 때 어떤 서비스가 반응하지 않도록 하는데도 유용할 것이다.

- 프로세스 메모리 보기

만약 포맷 함수의 응답(출력 스트링)을 볼 수 있다면 그것으로부터 유용한 정보를 수집할 수 있으며(왜냐하면 그것은 우리가 통제하는 행위의 출력이기 때문에), 그리고 공격자는 이 결과를 포맷 스트링이 무엇을 하며, 프로세스 배치가 어떻게 생겼는지에 대한 개략적인 내용을 획득하는데 사용할 수 있다.

스택 보기 : 스택 메모리의 몇몇 부분을 볼 수 있다.
어떤 위치에서 메모리 보기 : 스택 메모리와 다른 메모리 위치를 보는 것도 가능하다. 이것을 위해 공격자는 포맷 함수로 하여금 우리가 제공할 수 있는 어떤 주소로부터 메모리를 보여주도록 해야 한다. 이것은 두 가지 문제를 제시한다. 첫 번째, 우리는 스택 파라미터로서 주소를 사용하고, 그곳으로부터 메모리를 보여주는 포맷 파라미터를 찾아야 하고, 그리고 그 주소를 제공해야 한다.

- 임의의 메모리 덮어쓰기

공격은 어떤 프로세스의 명령(instruction) 포인터의 통제권을 장악하는 것이다.

버퍼 오버플로우와 유사한 공격 : 포맷 스트링 취약점은 가끔 버퍼 길이 제한에 관한 방법을 제공하고, 일반적인 버퍼 오버플로우와 비슷한 공격을 허용한다.
순수 포맷 스트링을 통한 공격 : 만약 간단한 공격 방법을 적용할 가능한 방법을 가지고 있지 않다고 해도, 여전히 그 프로세스를 익스플로잇할 수 있다. 그렇게 함으로써 실제 실행 통제에 아주 제한된 통제를 확장하고, 그것은 시스템 코드를 실행시킬 수 있다.

o 포맷스트링 취약점 점검툴

일단 공격이 끝나거나 또는 exploit을 개발하는 도중이라도 필요한 offset을 저장하기 위해 툴을 사용하는 것은 도움이 될 것이다. 어떤 툴들은 소스가 공개되어 있지 않은 소스 소프트웨어에 존재하는 포맷 스트링 취약점과 같은 취약점들을 확인하는데도 도움이 될 것이다. 여기서 언급한 네 가지 툴은 아주 많은 도움이 될 것이다.

- ltrace, strace

ltrace와 strace는 비슷한 방식으로 작동하는데, 프로그램이 호출할 때 그들의 인자와 리턴값을 로깅하면서 라이브러리와 시스템 호출을 후크(hook)한다. 이것은 프로그램 그 자체를 블랙 박스로 간주하면서 어떻게 프로그램이 시스템과 상호 작용하는지를 우리가 볼 수 있도록 해준다. 이미 만들어진 모든 포맷 함수는 라이브러리 호출과 그들의 인자들이며, 가장 중요한 것은 그들의 주소들이 ltrace를 사용하여 확인될 수 있다는 것이다. 이런 식으로 한다면 우리가 ptrace할 수 있는 어떤 프로세스의 포맷 스트링 주소를 신속하게 확인할 수 있다. strace 프로그램은 데이터가 읽혀지는 버퍼의 주소를 확인하는데 사용된다. 이 두 가지 툴을 배우기 위해서는 많은 시간이 필요한데, 이 툴들을 GDB attach 하는데 사용할 것이다.

- GDB, objdump

고전적인 GNU 디버그인 GDB는 텍스트 기반의 디버그인데, 소스 차원과 머신 코드 디버깅을 위해 적절하다. 일단 이것에 익숙해지면 프로그램 인터널에 대한 강력한 인터페이스가 될 것이다. exploit을 디버깅하는 것으로부터 프로세스가 익스플로잇되는 것을 지켜보는 것까지 어떤 것을 위해서도 편리하다. Objdump은 메모리 레이아웃과 같은 실행 가능한 바이너리나 오브젝트 파일에 대한 정보를 알아내는데 적절한 프로그램이다. 우리는 주로 바이너리로부터 GOT 엔트리의 주소를 알아내는데 사용할 것이다. 물론 다른 유용한 방식으로도 사용될 수 있다.

3.2 보안 신기술

인터넷에서는 언제나 약점이 있고, 재미를 위해서 혹은 이익을 위해서 항상 문제를 만들어내는 사람도 존재하기 때문에 인터넷을 완전하게 보안한다는 것은 불가능한 일일 수 있다. 절대적인 보안이 가능한 어느곳에서든 현재의 상황을 향상시키기

위한 작업으로부터 분리시키려는 것은 불가능한 목표라는 사실을 인식해야 한다. 따라서 수많은 불가능한 단계들이 보다 안전한 환경을 위한 인터넷의 일부분으로써 발전하도록 형성해 가는 작업이 필요하다. 다음 내용들을 통해 계속해서 향상되고 진화중인 보안기술의 기본 개념과 방향을 이해해 두는 것이 좋다.

3.2.1 암호 알고리즘의 성능향상과 새로운 암호 알고리즘[1급]

(1) 암호 알고리즘의 성능 향상

o 암호 알고리즘의 안정성 강화
- 새로운 수학적 문제에 기반한 암호 알고리즘 개발
o 암호 알고리즘의 고속화
- RSA 암호 알고리즘의 고속화
- 타원곡선 알고리즘의 응용
o 구현의 용이성

(2) 새로운 암호 알고리즘

o 핵심가이드
- 암호 알고리즘의 종류와 주요 알고리즘에 대한 개념 이해
- 각 알고리즘이 제공하는 보안 서비스 종류 이해

(가) PGP (Pretty Good Privacy)

여러 가지 암호화 알고리즘들을 하나로 모아서 복합적으로 적절히 적용시키는 것이다. PGP 특유의 암호화 알고리즘은 존재하지 않는다. 공개키 암호화 방식은 네트워크 환경에 적합한 암호화 알고리즘이다.

암호화 하는데 많은 시간이 걸리기 때문에 PGP에세 메시지를 암호화할 때 실제로 관용 암호화 방식을 사용한다. 대신 공개키 암호화 방식은 메세지를 암호화하는데에 쓰이며 그 키를 상대방에게 보내는 목적으로 쓰이고 있다.

o PGP 기능
– 기밀성 : RSA(공개키 암호화 방식) + IDEA(관용 암호화 방식

(관용암호방식에 사용될 키를 공개키를 가지고 있는 송신자가 생성하여 메시지를 암호화 시킨 다음, 이때 사용된 IDEA키를 공개키로 암호화 하여 함께 보낸다)
- 메세지 인증 : MD5 해쉬함수의 128bit 해쉬값 이용한다.
- 메세지 압축 : ZIP 알고리즘 사용가능
- 전자우편과의 호환 : Padix-64 conversion을 이용
- Pass Phrase
- Fingprint

(나) S/MIME (Secure / Multipurpose Internet Mail Extension)

기존 전자우편 보안 시스템의 문제점인 PEM 구현의 복잡성, PGP의 낮은 보안성과 기존 시스템과의 통합이 용이하지 않는다는 점을 보완하기 위해 IETF의 작업 그룹에서 RSADSI의 기술을 기반으로 개발된 전자우편 보안시스템 이다.

o S/MIME의 메시지 구성
- Enveloped data
- Signed data
- Clear-signed data
- Signed and Enveloped data

(다) SSL (Secure Socket Layer)

SSL은 웹 브라우저 개발로 이미 잘 알려진 네스케이프 (Netscape) 사에서 처음으로 제안 되었으며, 자사의 웹 어플리케이션에 처음으로 구현함으로써 현재 웹 보안의 대명사로 알려져 있는 보안 프로토콜이다. 그러나 웹과 같은 특정 응용을 위한 보안 프로토콜이 아닌 일반적인 인터넷 보안 프로토콜로도 사용 될 수 있다. TCP/IP 계층과 어플리케이션 (HTTP, TELNET, FTP 등) 사이에 위치하여 데이터를 송신하는 두 컴퓨터 사이, End to End 보안 서비스를 제공한다.

o SSL 구조
- Handshake Protocol
- Change Cipher Spec
- Alert Protocol

- Record Protocol : 실질적인 보안 서비스 제공

o SSL에서 제공하는 보안 서비스

SSL은 상호 인증 (Mutual Authentication), 무결성을 위한 메시지 인증코드 (MAC : Message Authentication Code), 기밀성을 위한 암호화 등을 제공함으로써 클라이언트와 서버 사이에 안전한 데이터 통신을 제공한다. 또한 이 프로토콜은 암호화, 메시지 압축, 메시지 인증 코드 (MAC)을 위해 사용되는 알고리즘을 선택하는 것이 가능하다. 이렇게 함으로써 암호 제품 사용에 대한 법적인 문제, 수출입 등에 따른 제반 문제에 맞춰 특정 서버에서 암호 알고리즘을 선택할 수 있으며, 새로운 알고리즘을 쉽게 이용하는 것이 가능하다.

- 두 응용간의 기밀성 서비스 : DES 이용
- 클라이언트와 서버의 상호 인증 : RSA, DSS, X.509
- 메시지 무결성 서비스 : MAC

(라) TLS (Transport Layer Security)

두 어플리케이션간에서 메시지 기밀성, 데이터의 무결성을 제공하기 위한 프로토콜로써 TLS 레코드 프로토콜과 핸드쉐이크 프로토콜로 구성 되어 있다.

o TLS 목적

- 계층간 보안 : 통신 계체 사이의 연결성 보장

- 상호 호환성 : 암호화 파라메터들의 교환은 TLS가 담당하고 있으므로 개발자들이 상대방의 암호 코드를 모르는 상태에서 응용 프로그램을 개발할 수 있다.
- 확장성 : 새 프로토콜과 라이브러리를 만들 필요 없음
- 효율성 : 암호화, 특히 공개키 암호화시 CPU에 많은 부하가 발생하기 때문에 TLS 프로토콜은 생성되는 연결의 수를 줄이기 위해 세션 캐쉬 구조를 채용

o TLS 보안 서비스

- 전자서명 : 해쉬함수, RSA, DSS

– 스트림 암호화 : 의사난수 생성기에서 생성된 값과 XOR 암호화
– 블록 암호화 : CBC (Cipher Bolck Chaining) 이용
– 공개키 암호화 : RSA 암호화 값은 PKCS#1 block type 2에 의해 암호화

3.2.2 새로운 인증기술[1급]

o 핵심가이드
- 생체인증 시스템의 개념 이해
- 생체인증의 종류

(1) 생체인증 시스템

자신의 신체 혹은 행동 특성을 등록시키고 나면, 보안을 요구하는 모든 곳을 자유롭게 이용할 수 있게 된다. 어떤 사람이 요구하는 ID를 인증하기 위해서는, 그 사람이 바이오메트릭 시스템에 등록 시 저장된 원본 데이터와 새로이 제시한 새로운 샘플을 비교한다. 이때 원본 데이터 샘플은 그들이 항상 가지고 다니는 스마트 같은 메모리에 보관될 수 있다.

(2) 생체 인증의 종류

o 신체특성 이용
- 지문 인식
- 얼굴 인식
- 손의 형태 인식
- 홍채 및 망막 인식
- 저액 패턴 인식
- 귀 인식
- 입술 인식

o 행동학적 특성 이용
- 서명 인식
- 음성 인식
- 걸음걸이 인식

- 키 스트로크 인식

(3) 생체인증 시스템의 동작 절차

o 등록
o 인증
o 안정성 여부 판별
o 오인식율 (FAR) 과 오거부율 (FRR)
o 바이오메트릭의 스마트 카드 적용

(4) 생체인증 기반의 전자서명

o 전자펜으로 입력된 서명의 진위를 판별하는 기술
o 영상 인식 기술

(5) 타기술과 PKI의 접목을 통한 인증방법의 강화

o 생체인증과 PKI
- 공인인증서와 생체정보를 결합한 편리하고 완벽한 사용자 인증 기능 지원
- 단순한 지문 이미지가 아닌 생체정보를 이용한 알고리즘
- 암호화 기술에 의한 안전한 생체정보의 전달 및 관리기능

o 주요 서비스 분야
- 인터넷 뱅킹
- 사이버 증권
- 쇼핑몰에서 상품주문 및 지불결제
- 인터넷 유료 컨텐츠 이용
- 무선 인터넷

(6) 차세대 네트워크를 위한 PKI

개방형 네트워크 또는 분산형 네트워크 환경에서 보안 요구사항을 만족시키기 위해서는 공개키 암호와 인증서의 사용을 가능하게 해주는 새로운 기반구조가 필요하

게 되는데, 이러한 공개키 암호기술을 이용한 기반구조를 공개키 기반구조 (PKI) 라고 한다.

o 핵심가이드

- 차세대 PKI 기술의 기술 동향의 이해
- WPKI의 인증서 발급 절차 이해

o 무선 PKI 기술

- 무선 PKI 구축 기술
- 무선 단말의 제한된 리소스에 따른 암호 알고리즘

o 유, 무선 PKI의 통합 기술

- 차세대 네트워크 신기술과 PKI의 통합

o WPKI 인증서 발급 절차

WPKI 인증서 발급 절차는 발급받는 WPKI 클라이언트가 단말인 경우 발급정보를 보호하고 POP (Proof Of Possession)의 확인하는 절차에 초점이 맞춰진다.

인증서 발급 절차는 WAP Forum에서 제시하고 있는 것과 국내 표준과 다르며, 국내 표준은 발급 정보 보호 및 POP 확인을 WTLS 대신 일회성 Passwordbased MAC과 SignText를 이용하도록 되어 있다

- CA -> 단말 : nonce 전달
- 단말 : nonce를 이용 계산
- 단말 -> CA(RA) : SignedContent 전달
- CA : SignedContent에서 M 출력

o Global PKI

무선 사용자에게 정보보호 서비스를 제공하는 무선 PKI를 기존 유선 PKI 시스템과 연동한다. 또한 OCSP, SCVP, DVCS, TSP 등과 같이 PKI 고도화 기능을 통해 실시간 거래, 고액 거래, 내용 증명 등과 같은 정보보호 서비스를 제공하며 권한관리 기반 구조인 PMI를 통해서 사용자의 접근 제어 관리를 수행하는 시스템을 말한다.

(7) RFID 보안

o 핵심가이드

- RFID의 개념 이해
- RFID 보안상 취약점과 프라이버시 보호기술의 종류 이해

o RFID (Radio Frequency Identificattion)

마이크로 칩을 내장한 태그, 라벨, 카드 등에 저장된 데이터를 무선 주파수를 이용하여 리더기에서 자동 인식하는 기술을 말한다.

o RFID 시스템

- RF 리더 : 판독 해독 기능
- RF 전자태그 (transponder) : 정보 제공 역할
  a. 전력에 따른 분류
  b. 칩의 유무에 따른 분류
  c. Read-only와 Read Write
  d. 주파수에 따른 분류 : UHF, HF, LF
- RRID 태그 식별 프로토콜
  a. anti-collision 프로토콜 수행
  b. Singulation 프로토콜 수행
  c. ALOHA 프로토콜 : HF 태그의 충돌 방지

o RFID 보안 및 프라이버시 노출 위협요인

- RFID 제약조건 : 암호화를 수행하기 위한 칩에 가격문제
- 관련 공격기법 및 위험요인 : EPIC에서 분석
  a. 숨겨진 태그 장소
  b. 유일한 식별자
  c. 대규모 데이터 통합
  d. 숨어있는 리더
  e. 개인 추억과 개인정보 프로파일
- 위협요인
  a. 도청공격
  b. 트래픽 분석
  c. 스푸핑

d. 서비스거부
e. 세션 가로채기
f. 물리적 공격

o RFID 보안 및 프라이버시 보호 기술
- AutoID센터의 Kill Tag 기술
- Faraday Cage 기술
- Active Jamming 기술
- MIT의 해쉬-락 기술

o RFID 산업화 핫 잇슈
- RFID 가격 최소화
- 극소화
- Read/Wirte 시스템
- Network간의 호환성 확보
- 주파수 확보
- 프라이버시 침해방지
- 센서네드워크 및 무선기술 공존 활용

3.2.3 DRM[1급]

각종 콘텐츠의 디지털화와 인터넷 인프라의 확산에 따라 부가가치가 급상승 하고 있는 디지털콘텐츠의 보호는 매우 중요한 영역으로 자리매김하고 있다. 이런 요구에 의해 탄생한 DRM(Digital Rights Management) 즉 "디지털 저작권(권리) 관리" 기술과 관련 정보보호 제품은 특히 앞으로 전개될 유비쿼터스 환경 하에서 매우 중요한 요소가 아닐 수 없다. 본 출제영역에서는 DRM의 개념과 특징을 이해하고 그 적용분야에 대한 지식을 요구한다.

o 핵심가이드
- DRM의 정의와 특징 이해
- DRM의 기술구성요소와 응용분야에 대한 이해

(1) DRM의 개요

o DRM의 정의

DRM(Digital Rights Management)기술은 콘텐츠의 지적재산권이 디지털 방식에 의해서 안전하게 보호, 유지되도록 콘텐츠 장착에서부터 소비에 이르는 모든 유통 과정에서 거래 및 분배규칙, 사용규칙이 적법하게 성취되도록 하는 기술이다.

o DRM의 구성요소

- 콘텐츠(Contents) : 지적자산의 가치가 있는 정보 단위이며, 허가되지 않은 사용자로부터 보호해야 할 대상
- 사용자(User) : 부여된 접근권한(Permission)과 상태(condition)에 따라 콘텐츠를 이용할 주체로 콘텐츠의 생산자, 배포자, 사용자가 될 수 있다.
- 접근권한(Permission) : 콘텐츠의 이용권리는 콘텐츠별로 정해진 퍼미션에 의해 결정된다.
- 상태(Condition) : 접근권한이 수행되기 위한 요구조건 및 제한요소를 포함한다.

o DRM의 시스템 구성요소

- 패키저(Packager) : 보호대상인 콘텐츠를 메타데이터와 함께 배포 가능한 단위로 패키징한다.
- DRM 컨트롤러(Controller) : 배포된 콘텐츠를 사용자의 플랫폼에서 콘텐츠의 이용권한을 통제한다.
- 클리어링 하우스(Clearing House) : 콘텐츠에 대한 배포 정책 및 라이선스를 발급, 관리한다.

o DRM의 주요기술

저작권보호를 위한 워터마킹기술, 콘텐츠를 패키지하여 라이센스 메커니즘을 통해 구매자에게 콘텐츠를 전달하는 유통기술, 콘텐츠별 응용 서비스 기술 등이 디지털 컨텐츠 유통기술의 핵심이며, 또한 저작권 정보 이외에 추가로 콘텐츠에 구매자 정보를 삽입하여 콘텐츠 불법 복제자를 추적하는 핑거프린팅 기술 등도 중요한 기술이다.

(2) DRM의 주요특징

o 용이한 사용자 최적화 및 연동성 제공
o 유동적인 서버 운용 및 CA 관리 기능
o 판매방식의 다양한 지원, CRM(Customer Resource Management) 서비스 제공
o 최고의 안정성을 가지는 국제 표준 암호화 알고리짐의 제공
o 대용량 전송 네트워크 지원
o 실시간 라이센스 키 다운로드 지원
o 다양한 형태의 멀티미디어 콘텐츠 지원
o 콘텐츠 복사 방지 및 사용횟수 제한 등의 복제방지관리(Copy control) 가능

(3) DRM의 적용분야

o AOD(Audio on Demand) 서비스
o VOD(Video on Demand) 서비스
o 웹 캐스팅 서비스 및 광고서비스
o 온라인 교육 콘텐츠 서비스
o 전자책(E-Book) 관련 콘텐츠 서비스
o 온라인 뱅킹 서비스

DRM이 가장 먼저 적용된 분야는 디지털 콘텐츠의 상거래 분야로 디지털뮤직, 동영상, 이미지, 전자책/만화, 그리고 기타 부가정보서비스 등으로 디지털 콘텐츠에 대한 불법사용을 방지할 목적으로 이용되어 왔다.

콘텐츠 유통분야의 경우는 그 적용에 따라 콘텐츠 제공업자는 Pay-to-Use, Preview-before-Purchase, Subscribe 등 다양한 마케팅 정책의 수행과 사용자 권한에 따라 보기(View)/실행(Play), 프린트, 편집(Edit), 저장(Save), 전송(Transfer) 등 세분화된 콘텐츠 이용통제가 가능하다.

DRM은 또한 기업 및 주요 기관에서 생산, 관리하는 중요 기밀문서의 보안에도 적용이 되고 있다.

(4) DRM 기술의 예

o PKI 기반의 불법복제방지 기술

- 개념 : 콘텐츠를 소비자의 암호화 키를 이용하여 패키징함으로써 이를 다른 사람들이 이용할 수 없도록 하는 방식을 사용한다.
- 단점 : 소비자에게 종속적인 암호화를 수행함에 따라 콘텐츠 배포 서버의 프로세스 부담이 가중되고 슈퍼 배포자와 같은 기능이 없어 디지털 콘텐츠 유통에는 적합하지 않다.

o DOI 기반의 저작권 보호기술

- 개념 : DOI(Digital Object Identifier)는 저작권 관리 정보를 바탕으로 저작권 인증을 부여하는 기술이다.
- 단점 : 불법복제 및 불법사용 방지 기능이 제공되지 않고 있어 적극적인 저작권 보호를 하지 못하고 있다.

(5) DRM의 표준화 및 향후전망

o 활동분야별 분류

국내에서는 산학연을 포괄하여 운영되는 DRM 워킹그룹(WG)을 통해 DRM 표준화가 추진되고 있으며, 한국전자통신연구원(ETRI)은 국내 표준 DRM 기술개발을 추진하고 있다. 국제적으로는 표준화 추진단체인 MPEG21(Moving Picture Experts Group21)이 가장 적극적이고 활발한 활동을 진행하고 있다. MPEG21 이외에도 TV-Anytime, W3C, DOI 등의 다양한 표준화 활동을 진행하고 있는 단체가 있다.

최근에는 기존에 콘텐츠 유통으로 집중되었던 DRM에서 지불과 관련된 연구가 진행되고 있으며, 콘텐츠 제작에 있어 DOI에 대한 표준 작업과 연계하여 진행되고 있다.

- DRM 플랫폼의 표준사양 개발분야 : MPEG-21, TV-Anytime, APP, OeBF, OMA
- 식별체계의 표준화 기술개발 분야 : DOI, URI, MPEG-21 DII
- 권리표현기술 개발분야 : XrML, ODRL, XMCL, MPEG-21 RDD/REL
- 지적자산의 메타데이터 정보관리 기술분야 : INDECS, ONIX, 더블린코어

o DRM 향후 전망

DRM이 상용화 된 것은 2000년 초반부터이다. 현재의 콘텐츠 유통 영역이나 기업의 문서보안 영역 외에 다양한 응용영역으로 활용될 수 있으며 DRM 기술은 장기적으로 다음과 같은 방향으로 발전할 것으로 전망하고 있다.

- 벤더 및 특정 기관에 의해 분산돼 개발되던 각종 기술들이 표준화되어 유기적으로 통합된 플랫폼으로 발전
- 유비쿼터스 환경에서 디지털 콘텐츠의 투명한 접속 및 이용, 콘텐츠의 사용권리의 자유로운 이동 등 사용 편리성이 충분히 보장되는 방향으로 발전
- DRM 기술이 운용체계에 기본사양으로 포함될 것이다.
- 디지털의 유통구조가 대형화, 국제화 됨에 따라 이를 체계적으로 관리하고 투명한 유통 환경을 지원하기 위한 “클리어링 하우스 센터”가 구축될 것이다.

참고문헌

[1] William Stallings, *컴퓨터통신보안*, 그린, 2001년(Second Edition)

[2] 홍석범, *리눅스 서버보안 관리 실무*, (주)슈퍼유저코리아, 2005년 4월

[3] Network Working Group, *RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1*, IETF

[4] 한국정보보호진흥원, *홈페이지 개발 보안 가이드*, 한국정보보호진흥원, 2005. 4.

[5] OWASP, *OWASP TOP10*, OWASP, 2004. 7.

[6] Paul DuBois, *MySQL의 사용, 관리, 프로그래밍을 위한 완벽 가이드*, SAMS

[7] Informix Press, *INFORMIX Self Study Guide.*

[8] 한국Microsoft, *http://www.microsoft.com/korea/technet/*

[9] John Heimann, *ORACLE 9i APPLICATION SERVER VERSION 2의 보안 기능*, Oracle

[10] 한국전자통신연구원, 암호*학의 기초*, ETRI

[11] 김종필외 공저, *정보보호핵심지식*, 도서출판정일, 2004

[12] Ronald L. Hrutz /Russell Dean Vines, *The CISSP Prep Guide*, 사이버텍미디어, 2005(Second Edition)

[13] 한국전자거래진흥원, *e-Business(B2B) 표준기술적용 지침*, 한국전자거래진흥원, 2005.05

[14] 한국전자거래 진흥원, *http://www.kiec.or.kr*, ebXML

[15] 한국전자거래 협회/기술협회, http://www.kcals.or.kr// *B2B 표준화 가이드라인(v3.0)*, 2005.05

[16] 국가정보원, 정보통신부, *2005 국가정보보호백서*, 2005.06

[17] 한국HP, *IT리더양성과정,* 2005.11

[18] CCC Congress, *Exploit Format String Vulnerabilities,* 2001.09

[19] 에듀위즈(주), *Cryptography & Digital Signature,* 2005.05

[20] 에듀위즈(주), *Web Hacking,* 2005.10

[21] 에듀위즈(주), *Information System Security Technology,* 2005.09

[22] 에듀위즈(주), *Network Security Technology,* 2005.09

[23] 에듀위즈(주), *Current Hacking & Defense,* 2005.10

[24] 에듀위즈(주), *CISSP & CISA,* 2005.11