DNS 서버의 구분
1) Resursive(또는 Cache) DNS 서버 : 클라이언트 요청에 대해서 자신의 캐시에 데이터가 있거나 없으면 root dns 서버부터 계층적으로 질의하여 그 결과를 클라이언트에게 응답해주는 DNS 서버
2) Authoritative DNS 서버 : 위임받은(관리하는) 도메인에 대해서만 응답을 해주는 DNS 서버
윈도우 로컬 DNS 캐시 관련 명령어
ipconfig /displaydns : 로컬 DNS 캐시 정보를 조회 ipconfig /flushdns : 로컬 DNS 정보를 삭제
HTTP 주요 상태 코드
1xx : Information 100 - Continue(클라이언트로부터 일부 요청을 받았으며 나머지 정보를 계속 요청함)
2xx : Success 200 - OK(요청이 성공적으로 수행되었음) 201 - Created(PUT 메소드에 의해 원격지 서버에 파일이 생성됨 202 - Accepted(웹서버가 명령을 수신함)
3xx : Redirection(재지정 응답 코드, 요청 자원의 위치가 재지정 되었음을 의미) 301 - Moved Permanently(요청 자원의 위치가 영구적으로 변경됨, Location 헤더 필드를 통해 변경된 URL 응답) 302 - Found(요청 자원의 위치가 임시로 변경됨, Location 헤더 필드를 통해 변경된 URL 응답) 304 - Not Modified(요청한 자원이 변경되지 않았으므로 클라이언트 로컬 캐시에 저장되 자원을 이용하라는 의미)
4xx : Client Error 400 - Bad Request(요청 메시지 문법 오류) 401 - Unauthrized(요청한 자원에 대한 인가 필요, 요청 자원을 실행하는데 필요한 적절한 권한이 없음을 의미) 403 - Forbidden(요청한 자원에 대한 접근 차단) 404 - Not Found(요청한 자원이 존재하지 않음 408 - Request Timeout(요청 대기 시간 초과)
5xx : Server Error 500 - Internal Server Error(내부 서버 오류) 503 - Service Unavailable(일시적인 과부하 또는 서비스 중단 상태) 504 - Gateway Timeout(과부하 등으로 HTTP 요청에 대한 프록시(또는 게이트웨이)의 응답 대기 시간 초과
HTTP Method의 종류
1) GET : HTTP 요청 URL의 끝에 데이터(파라미터)를 추가해서 전달하는 방식
2) HEAD
3) POST : HTTP 요청 메시지 바디부에 데이터를 담아서 전달하는 방식 -> 히스토리나 서버의 access로그에 데이터가 남지 않음
4) OPTIONS : 웹서버에서 지원할 수 있는 메소드 목록을 담고 있는 헤더 -> 정보 자체가 취약점이 될 수 있으므로 OPTIONS 메소드는 허용하지 않도록 설정
5) TRACE
HTTP 요청 헤더의 의미
1) Referer 요청 헤더 - 링크를 통해 자원(페이지)에 접근할 때 해당 링크가 있는 이전(페이지)의 URL을 의미
2) Cache-Control 요청헤더 - max-age=0의 옵션 : 캐시 서버에 저장된 요청 자원의 유효성 유무와 무관하게 원본 웹서버로 유효성 검증을 다시 하라는 의미 -> 캐시 서버를 무력화시켜 원본 웹서버에 부하를 발생시키는 공격에 악용
TFTP 서비스의 Secure mode
- chroot 기능을 이용하는 것으로 지정한 디렉터리를 최상위 디렉터리로 지정하여 지정한 디렉터리의 상위 디렉터리로 접근하지 못하도록 제한하는 방식
설정 -> '-s 최상위 디렉터리' 형식으로 실행 인수를 지정해서 설정한다.
로그관리 관련 프로토콜
syslog : 다양한 장비의 로그 메시지를 원격 로그 수집기로 보내 로깅과 분석을 수행하기 위한 UDP기반의 인터넷 표준 프로토콜
SNMP : IP 네트워크상의 장비로부터 정보를 수집 및 관리하며, 정보를 수정하여 장치의 동작을 변경하는 데 사용되는 인터넷 프로토콜
SNMP
네트워크상 호스트들의 관리 정보를 수집하거나 실시간 모니터링 및 설정을 제공하는 프로토콜

<구성 요소>
- 매니저 : 관리 시스템 / 에이전트에 필요한 정보를 요청하는 모듈
- 에니전트 : 관리 대상 / 필요한 정보를 수집해서 매니저에게 전달해주는 역할을 하는 모듈

<데이터 수집 방식>
- 폴링 방식 : 161/upd, 매니저가 에이전트에게 정보를 요청하면 응답해주는 방식
- 이벤트 리포팅 방식 : 162/udp, 에이전트가 이벤트 발생 시 이를 매니저에게 알리는 방식

SNMP 서비스 사용시 고려해야 할 보안 설정

1) SNMP 서비스를 사용하지 않는 경우 비활성화하여 불필요한 정보 유출 및 불법 수정 등이 발생하지 않도록 한다

2) SNMP 서비스를 사용 시에는 데이터 인증 및 암호화 등 보안 서비스를 제공하는 SNMPv3를 사용한다

3) SNMP Community String은 SNMP 서버와 클라이언트가 데이터를 교환하기 전에 인증을 위해 사용하는 일종의 패스워드로 초기값이 public 또는 private로 설정되어 있다. 이를 유추하기 어려운 복잡한 문자열로 변경

4) SNMP 서비스는 RO(Read Only)와 RW(Read Write)모드를 제공한다. 쓰기 권한이 있으면 시스템 중요설정을 수정하는 등 보안 문제를 유발할 수 있으므로 반드시 필요한 경우가 아니라면 쓰기 권한 사용을 자제한다.

SNMPv3의 다양한 공격 위협으로부터 방어하기 위한 매개변수

1) 재전송 공격 방지 -> SNMP 엔진 ID, 부트 횟수, 엔진 시각 정보를 이용하여 메시지의 유효시간을 계산, 재전송 여부를 판단
(매개변수)
- Authoritative 엔진ID(msgAuthoritativeEngineID)
- Authoritative 엔진부트횟수(msgAuthoritativeEngineBoots)
- Authoritative 엔진시각(msgAuthoritativeEngineTime)

2) 위장 공격, 메시지 위변조 공격 방지 -> 메시지 인증을 위해 HMAC(MD5, SHA)을 사용
(매개 변수)
- 사용자(msgUserName)
- 인증 매개변수(msgAuthenticationParameters)

3) 도청/스니핑(정보노출) 공격 방지 -> 메시지 암호화(DES-CBC)를 지원
(매개 변수)
- 암호 매개변수(msgPrivacyParameters)

SNMP 관련 용어

1) MIB(Management Information Base) 관리 정보 베이스
- SNMP가 사용되면서 관리자가 조회하거나 설정할 수 있도록 대행자에 의해 유지되는 변수를 정의
- 관리되어야 할 특정한 정보, 자원을 객체라고 하고 이러한 객체들을 모아 놓은 집합체를 MIB라한다
- 즉, 관리자가 조회하거나 설정할 수 있는 객체들의 데이터베이스

2) SMI(Structure Management Information)
- MIB를 정의하기 위한 일반적인 구조
- ASN.1 언어를 사용하여 데이터와 데이터의 속성들을 정의, 메시지 전송 시 비트 변환 규칙으로 ASN.1의 인코딩 룰 중 BER(Basic Encoding Rules)을 사용한다

크로스 사이트 스크립트(XSS)

공격자가 특정 웹 페이지에 실행할 수 있는 스크립트를 삽입하여 페이지 방문자의 개인정보나 쿠기 정보를 탈취하거나, 웹 사이트를 변조하거나, 악의적인 사이트로 사용자를 리다이렉트하는 공격

<대응 방안>
- 사용자 입력값은 반드시 서버 단에서 검증하여 악의적인 행위를 발생시키는 스크립트가 웹 프록시 등의 도구를 통해 우회하여 입력되는 것을 방지
- 사용자가 입력한 문자열에서 HTML 코드로 인식될 수 있는 특수 문자를 일반문자로 치환하여 처리(이스케이프 처리)한다 / 치환이 필요한 특수문자들은 <,>,&,",'등이 있으며 문자 변환함수를 사용하여 치환하여 저장하도록 한다
- 게시판 등에서 HTML 태그 허용 시 HTML 태그의 허용 리스트(White List)를 선정한 후, 해당 태그만 허용하는 방식을 적용

크로스 사이트 요청 위조(CSRF) 취약점

웹 애플리케이션에서 정상적인 경로를 통한 요청과 비정상적인 경로를 통한 요청을 서버가 구분하기 못할 경우 공격자가 스크립트 구문을 이용하여 정상적인 사용자가 조작된 요청을 전송하도록 함

<대응 방안>
- 사용자 접근권한 정보가 포함된 임의의 토큰(CSRF 토큰)을 서비스 요청 전 단계에서 매번 새롭게 생성하여 발급하고 요청 시 함께 전달하도록 한 후 요청에 포함된 토큰과 발급 시 해당 사용자 세션 정보에 저장한 토큰을 비교하여 정상적인 경로를 통한 요청인지 검증

XSS와 CSRF의 차이점

XSS는 악성 스크립트가 클라이언트 측에서 실행되어 클라이언트 정보를 탈취하거나 악성코드를 다운로드하는 등의 클라이언트에서 악의적인 행위가 발생

CSRF는 악성 스크립트가 클라이언트 측에서 실행된다는 점은 유사하지만, 사용자가 의도하지 않은 조작된 요청을 서버로 보냄으로써 사용자 권한으로 서버의 사용자 정보를 수정하는 등의 서버에서 악의적 행위가 발생

OWASP관련 내용

1. Broken Aceess Control : 취약점 접근 통제
- 엑세스 제어 : 사용자가 권한을 벗어나 행동할 수 없도록 정책을 시행
- 취약시 : 사용자는 주어진 권한을 벗어나 모든 데이터를 무단으로 열람, 수정 혹은 삭제등의 행위로 이어질 수 있음

2. Cryptographic Failures : 암호학적 실패
- 적절한 암호화가 이루어지지 않으면 민감 데이터가 노출될 수 있음

3. Injection : 인젝션
- 신뢰할 수 없는 데이터가 명령어나 쿼리문을 일부분으로써 인터프리터로 보내질 때 취약점이 발생

4. Insecure Design : 안전하지 않은 설계
- 구현의 결함 -> 누락되거나 비효율적인 제어 설계로 표현

5. Security Misconfiguration : 보안 설정 오류

6. 취약하고 오래된 요소

7. 식별 및 인증 오류

8. 소프트웨어 및 데이터 무결성 오류

9. 보안 로깅 및 모니터링 실패

10. 서버측 요청 위조(SSRF)
- 웹 애플리케이션이 사용자가 제공한 URL의 유효성을 검사하지 않고 원격자원을 가져올때마다 발생

- SQL Injection 공격 : 데이터베이스와 연동된 웹 애플리케이션에서 공격자가 입력 폼 및 URL 입력란에 SQL문을 삽입하여 DB로부터 정보를 열람(또는 조작)할 수 있는 공격 방법

수 있는 공격 방법

- 인증 및 세션 관리 취약점 : 계정 토큰과 세션 토큰이 적절히 보호되고 있지 않다. / 공격자는 패스워드나 키, 세션, 쿠키, 기타 인증관련 토큰을 공격하여 인증을 우회하고 다른 사용자의 ID를 가장할 수 있다.

디렉터리 리스팅 취약점

가 노출될 수 있는 취약점

<대응 방안>
- 웹서버의 디렉터리 리스팅 기능을 비활성화
- 아파치 웹서버 : 설정파일(httpd.conf)의 모든 디렉터리(/)섹션의 Option 지시자에 Indexes 설정을 제거
- IIS 웹서버 : 웹 사이트 등록 정보에서 홈 디렉터리의 "디렉터리 검색"기능을 해제(체크 해제)한다.

Apache 웹서버 환경 설정 파일

httpd.conf : 웹서버 설정 관련 파일

srm.conf : 웹서버 자원 관련 파일

access.conf : 웹서버 보안 접근제어 관련 파일

Apache웹 서버 설정 파일(httpd.conf)

1) ServerRoot [디렉터리] : 웹서버가 설치된 최상위 디렉터리 경로를 설정

2) PidFile [경로] : 웹서버가 기동될 때 자신의 PID를 기록할 파일의 경로를 설정

3) Listen [포트번호] : 웹서버의 리스너/서비스 포트를 지정

4) DocumentRoot [디렉터리] : 웹 애플리케이션의 최상위 디렉터리를 지정

5) ServerTokens prod | min | OS | full ... : 웹서버의 HTTP 응답 시 헤더의 Server필드를 통해 제공할 웹서버, OS, 모듈 등의 정보 레벨을 지정 / 보안상 최소한의 정보만을 제공하는 Prod로 설정할 것을 권장

6) ServerSignature on | off : 웹서버에서 직접 생성하는 에러 메시지 등에 웹서버, OS, 모듈 등의 정보를 제공할지 여부를 지정한다 / 보안상 정보를 제공하지 않는 off 설정을 권장

7) ServerAdmin [메일 주소] : 클라이언트에 전달되는 에러 메시지에 포함될 관리자 메일 주소를 설정

8) User nobody, Group nobody : 서비스를 제공하는 서버 프로세스를 실행할 사용자와 그룹을 설정 / 보안을 위해 root가 아닌 제한된 구너한을 가진 nobody 또는 애플리케이션 계정 및 그룹을 지정하는 것을 권장

9) Timeout [시간(초)] : 클라이언트의 요청에 의해 서버와 연결이 된 후 클라이언트와 서버 간에 아무런 메시지가 발생하지 않을 동안의 대기 시간으로 초 단위로 지정한다

10) MacClients [개수] : 동시 연결이 가능한 클라이언트의 최대 개수를 설정

11) KeepAlive on | off : 클라이언트와 서버 프로세스간의 연결을 일정 조건에 따라 지속시켜 동일한 연결을 통해 동일 클라이언트의 다수 요청을 처리하는 기능의 사용 여부를 설정

12) MaxKeepAliveRequest [개수] : KeepAlive 옵션 사용 시 클라이언트와 서버 프로세스 간에 연결을 지속시키는 동안에 허용할 최대 요청 건수를 지정 / 해당 요청 건수를 초과하면 연결을 종료

13) KeepAlliveTimeout [초] : KeepAlive 옵션 사용시 클라이언트와 서버 프로세스 간에 연결을 지속시키는 동안에 클라이언트의 마지막 요청 이후 다음 요청을 대기하는 시간(초 단위)로 지정한다.
: 해당 시간 동안 클라이언트의 추가 요청이 발생하지 않으면 연결을 종료

14) DirectoryIndex [페이지] : 클라이언트가 디렉터리만 명시하여 요청할 때 이를 처리할 기본 페이지를 지정 / 여러 개의 파일을 지정하면 먼저 지정한 파일을 우선 검색하고, 없으면 다음 파일을 검색

15) ErrorLog [경로] : 웹서버의 에러 로그 파일의 경로를 지정

16) Options 지시자에 설정된 indexes 옵션을 삭제 -> 디렉터리 리스팅(인덱싱) 취약점 제거
에 설정된 FollowSymLinks 설정 삭제 -> 심볼릭 링크가 가능한 취약점을 제거 -> 허용할 경우 모든 파일 시스템에 접근할 수 있게 됨

17) LimitRequestBody 지시자 : HTTP 요청 메시지 바디부에 데이터를 담아서 전달할 수 있는 전체크기를 제한하기 위한 목적으로 사용

18) ORDER 지시자 : 뒤에 나오는 ALLOW나 DENY 항목부터 우선 적용(EX. Order deny,allow면 allow 항목부터 우선 적용

Apache 웹서버 로그 파일

1) 액세스 로그(access_log)
- 웹서버에 접속한 클라이언트의 지원(파일 등) 요청 내역을 기록해 놓은 로그
- 다른 프로그램으로 분석되고 요약될 수 있다
(주요 추출/분석 되는 정보)
- 홈페이지에 방문한 방문자 수
- 방문자들의 유형(방문자들이 접속된 서버의 도메인 이름으로 판단)
- 각 웹 페이지별로 얼마나 많이 요구되었는지 사용 시간대별 분석

2) 에러 로그(erroe log)
- 클라이언트 요청에 대한 웹버서 처리중에 발생하는 오류 및 디버깅 메시지 등을 남긴다

윈도우 시스템의 서비스별 로그파일 경로

IIS 서비스 관련 로그파일 경로
- 웹 서비스 로그 : %SystemRoot%system32₩Logfiles₩W3SVC1
- FTP 서비스 로그 : %SystemRoot%system32₩Logfiles₩MSFTPSVC1
- 웹 서비스 관련 오류 로그 : %SystemRoot%system32₩Logfiles₩HTTPERR

DHCP 서비스 관현 로그 파일 -> %SystemRoot%system32₩dhcp

DNS 서비스 관련 로그파일 -> %SystemRoot%system32₩dns

FTP 서비스에 대한 공격 유형

1) FTP bounce 공격
- 제어 채널과 데이터 채널을 다르게 사용하고 데이터 채널을 생성할 때 목적지를 확인하지 않는 FTP 설계의 구조적 취약점을 이용하는 공격
- 주로 익명 FTP 서버를 이용, PORT 명령을 조작하여 공격 대상 네트워크를 스캔하고 FTP 서버를 통해 공격자가 원하는 곳으로 데이터를 전송하게 한다

2) Anonymous FTP 공격
- 익명 FTP 서비스는 익명 계정으로 FTP 접속이 사능한 서비스를 말한다. 익명 계정을 허용하면 원격에서 인증 없이 누구나 서버의 파일에 접근할 수 있으므로 보안상 심각한 문제가 발생할 수 있다.
- 만약 익명 사용자에 쓰기 권한까지 있다면 악성코드를 업로드하여 다수의 사용자에게 피해를 발생시킬 수 있다

3) TFTP 공격
- 별도의 인증과정없이 지정된 디렉터리에 접근할 수 있는 보안상 취약점이 존재

SQL Injection 관련

Prepared Statement(전처리 질의문)
- 일반적인 질의문(동적 쿼리 방식)은 요청 시마다 질의문을 생성하여 실행시키는 방식이지만 Prepared Statement는 외부로부터의 입력값(변경되는 값)을 제외한 질의문 부분을 미리 컴파일하여(질의문 분석 및 최적화) 생성한 후 반복적인 입력값만을 매개변수로 전달하여(바인딩하여)질의문을 실행하는 방식

<방지 대책>

SQL 질의문 자체에 영향을 미치지 않는다.

2) 데이터베이스와 연동하는 사용자 입력 파라미터를 서버단에서 점검하여 SQL 질의에 사용되는 특수 문자(',",;,--,#등) 및 예약어(select, update, insert, delete 등)를 필터링

<적합하지 않은 대응>

- 스크립트로 문자열 필터링 정책 : 클라이언트에서 동작하는 문자열 필터링 스크립트(자바스크립트)는 클라이언트 단에서 직접 필터링 코드를 제거하거나 웹 프록시와 같은 도구를 이용해 쉽게 우회할 수 있기에

Blind SQL Injection

<공격 원리>
조건절의 참/거짓 결과에 따른 응답의 차이점을 이용한 공격기법으로 참/거직 결과에 따른 웹서버의 응답을 통해 원하는 DB 데이터를 추출하는 원리

<공격 과정>
1) limit를 통해 출력 행을 하나씩 증가시키면서 BASE TABLE(사용자가 만든 테이블 타입) 타입의 테이블정보를 row하나씩 추출
2) 하나의 row에 대해 그 테이블 명을 알기 위해 테이블 명 컬럼을 substring을 통해 한글자씩 비교함으로써 테이블 명을 완성
3) 결과적으로 공격자는 Information_schema.table에 저장된 BASE TABLE 타입의 테이블 명을 알아낼 수 있음

- ascii(문자) : 문자에 해당하는 ascii 코드값을 반환한다.
- substr(문자열, pos, len) : 문자열을 pos부터 len만큼 잘라낸다
- limit off, len : 출력 레코드의 개수를 제한하는 함수, off는 시작 행을 의미하며 0부터 시작, len은 개수를 의미 -> off를 1씩 증가시키면 다음 레코드를 하나씩 순서대로 추출 가능
- information_schema : 데이터베이스에 대한 메타데이터를 제공하는 스키마
- information_schema.tables : 데이터베이스에 있는 모든 테이블 정보를 가지고 있는 테이블
- information_schema.table의 table_type : 테이블의 유형, 기본적으로 사용자 테이블을 생성하게 되면 테이블 타입이 base type으로 설정됨

웹 애플리케이션 불충분한 세션관리 취약점 점검 방법

검

2) 웹 세션 ID 중복 사용 점검 방법 : 로그인과 로그아웃을 반복하여 세션ID를 수집한 후 패턴을 분석하여 동일 세션 ID(또는 추측할 수 있는 패턴을 보인 세션 ID)가 반복적으로 사용된다면 취약함으로 판단

named.conf == 네임서버의 설정 파일

```
zone '관리 도메인(zone)이름 지정' IN {
    type 'master (타입 지정)';
    file "도메인 존 파일 지정";
    allow-transfer {슬레이브 네임서버 IP;}; --> 존전송을 허용
}
<존 설정>
zone '관리 도메인(zone)이름 지정' IN {
    type 'slave (타입 지정)';
    masters {마스터 네임서버의 IP;};   --> 존 전송을 요청하도록
    allow-transfer {none}; --> 존전송을 허용x
}
```

버전의 정보를 보여주지 않기위해 임의의 정보로 변경할 수 있음

데이터베이스 보안 위협

집성 : 낮은 보안등급의 정보를 조각별로 조합하여 높은 등급의 정보를 알아내는 방식으로 개별 항목의 정보보다 종합적인 정보의 보안등급이 높은 경우 심각한 문제가 발생한다

추론 : 보안등급이 없는 일반 사용자가 보안으로 분류되지 않은 정보에 정당하게 접근하여 기밀정보를 유추해 내는 행위

데이터 디들링 : 원본 정보 자체를 위/변조하거나 디스크 속에 대체할 자료를 만들어 두었다가 원본 정보에 끼워 넣거나 바꿔치기하는 방식

OAuth 2.0

인증을 위한 개방형 표준 프로토콜로 제3자 프로그램에 리소스 소유자를 대신하여 리소스 서버에서 제공하는 자원에 대한 접근권한을 위임하는 방식을 제공

인증(Authentication) : 클라이언트 신원을 검증하는 단계

인가(Authorization) : 자원에 접근할 권한을 부여하는 단계로 인가가 완료되면 리소스 접근권한이 담긴 Access Token이 발급된다

접근 토큰(Access Token) : 리소스 서버로부터 리소스 소유자의 보호된 자원을 획득할 때 사용하는 만료 기간이 있는 토큰 == 임의의 문자열로 보호된 리소스에 일정 기간 접근할 수 있는 권한을 가짐

갱신 토큰(Refresh Token) : Access Token 만료 시 이를 갱신하기 위한 용도로 사용하는 토큰

hosts.ics파일

웹 브라우저의 URL에 대한 DNS 질의순서
로컬시스템에 저장된 DNS Cache 정보 -> hosts.ics 파일 -> hosts 파일 -> DNS 서버 질의

일반적으로 사용하지 않는 인터넷 연결 공유 시 특정 클라이언트의 IP를 강제로 지정하는 기능을 하는 파일
-> host 파일을 굳이 변경안하고 hosts.ics파일을 변조 또는 생성하여 파밍사이트로의 유도할 수 있음

이메일 시스템의 구성요소

1) MTA(Mail Transfer Agent)
- 메일 서버 프로그램으로 수신한 메일을 분석하여 수신자가 자신이 아닌 메일 주소라면 해당 주소의 메일 서버로 전송하고(메일 릴레이 기능) 수신자가 자신의 메일 주소라면 MDA를 통해 각 사용자 메일함에 저장하도록함
- 대표적으로 샌드메일(Sendmail), 큐메일(Qmail)등의 프로그램이 있다

2) MDA(Mail Delivery Agent)
- 사용자 메일함에 메일을 저장해주는 프로그램으로 메일이 최종 수신 메일 서버에 도착했을 때 메일 서버는 MDA에 메일을 전달하고 MDA는 해당 사용자 메일함에 메일을 저장
- 대표적으로 프록메일(Procmail)이 있다

3) MUA(Mail User Agent)
- 사용자가 메일을 송수신하기 위해 사용하는 메일 클라이언트 프로그램으로 메일을 작성하거나 메일함에 도착한 메일을 확인하는 기능을 수행
- 대표적으로 아웃룩, 선더버드 등의 프로그램이 있다.

4) MRA(Mail Retrieval Agent)
- 메일 클라이언트(MUA)가 확인을 요청하는 메일을 사용자 메일함에서 사용자로 전달해주는 프로그램으로 MUA와 MRA간 통신은 POP3 또는 IMAP 프로토콜을 기반으로 한다

데이터베이스 보안 통제

1) 접근 통제 : 인증된 사용자에게 허가된 범위 내에서 데이터베이스 시스템 내부의 정보에 대한 접근을 허용하는 기술적 방법

2) 추론 통제 : 통계적인 자료에서 주로 발생하는 것으로 간접적으로 노출된 데이터를 통해 다른 데이터가 공개되는 것을 제어하는 방법

3) 흐름 통제 : 정보의 흐름으로 볼 때 보안 등급이 높은 객체에서 보안등급이 낮은 객체로 정보가 흐르는 것을 제어하는 방법

데이터베이스 보안 요구사항

1) 부적절한 접근방지(또는 접근 통제) : 비인가자에 의한 데이터 접근 차단

2) 추론 방지 : 기밀이 아닌 데이터로부터 기밀 정보를 얻어낼 가능성 차단

3) 데이터 무결성 : 비인가자에 의한 데이터의 생성, 변경 및 삭제 등으로부터 보호

4) 감사 기능 : 모든 데이터 접근에 대한 감사 기록(로그) 생성

5) 사용자 인증 : 데이터에 접근하는 사용자에 대한 신뢰성 있는 신원확인

데이터베이스 암호화 방식

1) API 방식 : 응용프로그램 자체 암호화 방식
- 애플리케이션(응용프로그램) 소스를 수정(암복화 API 적용)하여 암복화를 수행하는 방식
(장점) : 암복화가 애플리케이션 서버에서 발생하므로 DBMS 부하가 적고 애플리케이션 서버와 DB서버 간에 암호화된 데이터 전송 가능
(단점) : 애플리케이션 수정에 따른 비용 발생

2) Plug-in 방식 : DB 서버 암호화 방식
- DB서버에 플러그인 형태의 암복호화 모듈을 적용하여 암복호화를 수행하는 방식
(장점) : 애플리케이션 수정 불필요 또는 최소화
(단점) : DB 서버에서 암복호화를 처리하므로 대용량 처리 시 DB 서버 부하 발생

3) Hybrid 방식 : API 방식과 Plug-In 방식 혼용

4) TDE(Teansparent Data Encryption) 방식 : DBMS 자체 암호화 방식
- DBMS에 내장된 암복호화 기능을 이용하는 방식
(장점) 애플리케이션 수정이 불필요하고 DBMS 커널 수준에서 암복호화가 처리되므로 최적화된 성능을 제공
(단점) DBMS 종류 및 버전에 따라 기능 지원 여부가 결정

5) 파일 암호화 방식 : 운영체제 암호화 방식
- 운영체제(OS)에서 DB 파일을 직접 암호화하는 방식

MySQL관련

skip-networking 옵션 : 설정파일인 my.cnf의 항목에 추가함 -> 외부 네트워크 사용자의 접근을 제한하고 로컬에 설치된 애플리케이션에 의해서만 접근하도록 보안 설정

mysqldump : MySQL 데이터베이스 서버를 중단하지 않고 데이터를 백업할 수 있는 도구
MySQL 데이터 백업 방식
- 핫 백업 : 데이터베이스 서버를 중단하지 않고 온라인 상태로 유지한 채 백업하는 방식
- 콜드 백업 : 데이터베이스 서버를 중지한 후 데이터를 백업하는 방식

DNS Spoofing

희생자가 DNS질의를 수행하면 공격자가 이를 스니핑하고 있다가 희생자에게 조작된 DNS 응답을 보내 가짜 사이트로 접속을 유도하는 공격기법

<대응 방안>
- 중요한 사이트의 ip주소에 대해서는 DNS 질의보다 우선순위가 높은 hosts 파일을 통해 관리

DNS Spoofing의 구분

1) 스니핑 기반의 DNS 스푸핑 공격
- 희생자가 DNS 질의를 수행하면 공격자가 이를 스니핑하고 있다가 정상 응답보다 빠르게 희생자에게 조작된 IP주소를 담은 응답을 보내 조작된 주소로 접속하게 만드는 공격

2) DNS 캐시 포이즈닝(Cache Poisoning) 공격
- DNS 서버(일반적으로 캐시 DNS 서버)의 캐시 정보를 공격자가 조작하여 사용자의 DNS 질의 시 캐시에 저장된 조작된 주소로 응답하여 의도하지 않은 사이트로 접속하게 만드는 공격

vsFTP

1) anonymous 계정 접근제한
- anonymous 로그인 명(계정명)과 임의의 비밀번호를 사용하여 ftp 접근이 허용되는지 확인
- vsFTP 설정 파일(vsftpd.conf)에 anonymous_enable=NO로 설정한다

2) root계정 접근제한
- ftpusers 파일에 root 계정이 등록되어 있는지 확인 if) root계정이 등록되어 있지 않으면 root계정으로 접근이 허용됨
- vsftpd.conf파일에서 userlist_enable=YES인 경우, user_list 또는 ftpusers 파일에 root계정을 추가하여 접근을 제한
  userlist_enable=NO인 경우, ftpusers 파일에 root 계정을 추가하여 접근을 제한

ProFTP

1) anonymous 계정 접근제한
- proftp 설정 파일(/etc/proftpd.conf)의 anonymous 관련 설정 중 User, UserAlias 항목을 주석 처리한 후 서비스 재시작

2) root계정 접근제한
- vi 편집기를 이용하여 proftpd 설정 파일을 열기
- Root Login off 설정

XXE(XML eXternal Entity) Injection

XML 입력값을 파싱하는 애플리케이션의 취약점을 이용한 공격으로 악의적인 XML 공격 코드를 애플리케이션 입력값으로 전달(주입)하여 시스템 파일 접근이나 외부 파일의 참조 등 악의적인 행위수행

<동작 원리>
- 애플리케이션에서 XML 파서의 외부 엔티티 참조를 허용하는 경우, 시스템 파일을 참조하는 외부 엔티티를 선언한 후 이를 XML 내부에서 참조하는 방식으로 악의적인 시스템 파일접근이 가능하다.
- 이를 하나의 엔티티가 상위에 선언된 엔티티를 반복적으로 참조하게하여 부하를 발생시키는 일종의 도스 공격으로도 활용될 수 있다.

window DNS 서버설정

존 설정 : 관리하는 도메인을 DNS 서버에 등록

리소스 레코드 설정 : DNS 서버에 서비스 정보를 입력

DNS 주요 레코드

1) A 또는 AAAA : 도메인의 IP 정보, A : IPv4, AAAA : IPv6

2) NS : 도메인의 네임서버 정보

3) MX : 도메인의 메일 서버 정보

4) SOA : State Of Authority, 도메인의 기본 속성 정보(존 파일 버전, 전송 주기 등)

5) HINFO : Host Information, 도메인의 호스트 정보(CPU, OS 등)

6) CNAME : 도메인의 기준 도메인 정보 / 하나의 도메인(기준 도메인)에 여러 서브 도메인(별칭)을 부여하기 위해 사용

7) TXT : 도메인에 대한 텍스트 정보(대표적으로 SPF 레코드 정보가 있음)

8) PTR : IP에 대한 도메인 정보

DNS가 UDP 53번 포트가 아닌 TCP 53번 포트를 이용할 경우 이유

1) 존 전송 시 : 마스터에 있는 원본 존 데이터를 슬레이브가 동기화/복사하는 방법을 존 전송이라하는데 일반적으로 많은 존 데이터를 이동해야 하므로 신뢰성 있는 전송을 보장하는 TCP를 이용

2) 응답 데이터가 512 byte를 초과할 때 사용

Sendmail 관련 주요 파일 및 디렉터리

1) /usr/sbin/sendmail : sendmail 주 데몬 실행파일

2) /usr/sbin/makemap : sendmail 맵 생성 실행파일

3) /usr/spool/mqueue : 메일 큐 디렉터리(수신한 메일을 임시 저장하는 디렉터리)

4) /usr/spool/mail : 개별 사용자 메일함(개별 사용자별로 수신된 메일을 보관하는 디렉터리)

5) /etc/mail/access : 스팸메일 및 릴레이 차단을 위한 설정 -> access 파일은 설정을 위한 텍스트 파일로 실제 정책을 적용하기 위해서는 makemap 명령을 이용하여 access.db 파일을 생성해야 한다

6) /etc/mail/sendmail.cf : sendmail 설정 파일

Sendmail 서버 소프트웨어의 스팸 메일 방지를 위한 메일릴레이 설정

access 파일 : 메일 릴레이 설정을 위한 sendmail 데이터베이스 파일형

<설정의 의미>
- OK : 조건 없이 모두 허용
- RELAY : 메일 릴레이를 허용
- DISCARD : 메일 모두 폐기
- REJECT : 모두 거부(거부 메시지를 보냄)

makemap 명령 : 텍스트 형식의 access 파일을 sendmail이 인식할 수 있는 데이터베이스 파일 형태로 변환해주는 명령어

스팸메일을 방지하기 위한 보안 도구

1) Procmail
- Sendmail.cf에 포함 시키거나 사용자 디렉터리에 forward 파일을 위치시켜 실행시키는 Plug-In 형식으로 쓰임
- 받은 메일에서 보낸 사람, 제목, 크기, 내용 등의 단일 또는 다중기준으로 메일 필터링이 가능

2) Sanitizer
- e-mail을 이용한 모든 공격에 대해 효과적으로 대응할 수 있도록 해주는 procmail ruleset 옵션을 통해 확장자를 이용한 필터링, MS Office 매크로에 대한 검사기능, 악성 매크로에 대한 score 기능, 감염된 메시지 보관 장소 설정 기능 등을

3) Inflex
- 메일 서버에서 로컬이나 외부로 나가는 e-mail에 대한 Inbound와 Outbound 정책을 세워 메일 필터링이 가능
- 첨부파일만 필터링 할 수 있다

4) SpamAssassin
- Apache 그룹을 통해 Open Source Project로 진행되고 있음
- Perl로 만들어졌으며 헤더 내용, White List와 Black List, RBL(Real-time Black List, 실시간 블랙리스트)등의 다양한 분류로 메일 필터링이 가능

스팸메일을 방지하기 위한 이메일 표준인증 기술

1) SPF(Sender Policy Framework) 기술
- 발신(발송) 메일서버 정보를 사전에 발신 도메인 DNS 서버에서 공개적으로 등록함으로써 수신자가 수신한 메일의 발신자 IP 주소가 발신자 메일주소 도메인(MAIL FROM 정보)의실제 메일서버 정보(IP 주소)와 일치하는 지 검사할 수 있도

2) DKIM(DomainKeys Identified Mail) 기술
- 발신자가 발신(발송) 메일의 메시지 헤더와 바디에 대한 디지털서명을 메일 헤더에 삽입하여 수신된 이메일이 위변조되지 않았는지 수신자가 검증할 수 있는 이메일 인증 기술
- 전자서명 주체와 키 공유 방법 : 발신자는 발신 메일의 메시지 헤더와 바디를 자신의 개인키로 전자서명한 후 서명값을 메일 헤더에 추가하여 전송
  : 수신자는 발신자 도메인의 DNS에 등록된 전자서명을 검증할 수 있는 정보(서명 알고리즘, 공개키)를 담고 있는 DKIM 레코드를 조회하여 서명검증을 수행

3) DMARC(Domain-based Message Authentication, Reporting and Conformance)
- SPF 인증 기술과 DKIM 인증 기술을 DMARC 인증정책을 통해 모두 적용하여 정당한 발신자인지와 메일의 위변조 여부를 검사하고 수신 측에서 정책 처리 결과(인증 실패)에 대한 DMARC 보고서를 발신자에게 전달함으로써 발신자가 메

subdomain takeover(서브도메인 탈취) 또는 subdomain hijacking(서브도메인 하이재킹) 취약점

회사/기관의 DNS에 서브 도메인으로 웹 호스팅, 클라우드 등의 서비스를 CNAME 레코드로 등록하여 사용 후 서비스 이용을 중지했을 때, DNS에 서브 도메인을 삭제하지 않으면 공격자가 해당 회사/기관에서 사용했던 웹 호스팅, 클라우드 등의 서비스 도메인을 재사용하여 피싱등의 악의적인 사이트를 해당 회사/기관의 서브 도메인으로 운영할 수 있는 취약점

경로 추적/조작(Path Traversal 또는 Directory Traversal)취약점

웹서버와 웹 애플리케이션의 파일 또는 디렉터리 접근이 통제되지 않아 웹서버 또는 웹 애플리케이션의 중요한 파일과 데이터에 대한 접근 및 실행이 허용되는 취약점

<공격 원리>
- URL 상에 경로를 조작할 수 있는 다양한 인코딩된 특수문자들을 포함하고 있으며 이를 이용하여 시스템 디렉터리에 위치한 명령어(cmd.exe)를 실행
- 특수문자 URL 인코딩 예) . / ₩ 뒤에 %2e, %2f, %5c
- 특수문자 더블 URL 인코딩 예) . / ₩ 뒤에 %252e, %252f, %255c
- UTF-8 유니코드 인코딩 예) . / ₩ 뒤에 %c0%2e, %c0%af, %c0%5c

<대응방안>
- 사용자가 임의로 접근할 수 있는 최상위 디렉터리를 웹 애플리케이션 루트 디렉터리로 제한하여 웹서버의 시스템 루트 디렉터리로 접근하지 못하게 제한
  -> 유닉스/리눅스 시스템 : chroot 기능을 통해 웹 애플리케이션 루트 디렉터리를 가상의 root디렉터리로 만들어 시스템 루트 디렉터리의 접근을 차단
  -> 윈도우 시스템 : 새로운 논리 드라이브를 만들고 해당 드라이브를 웹 애플리케이션 루트 디렉터리로 변경
- 웹 방화벽 등 보안장비를 이용하여 경로를 조작할 수 있는 다양한 인코딩된 특수문자들을 필터링하도록 설정
- 취약한 버전의 웹서버 및 웹 애플리케이션 서버는 최신 버전으로 업데이트를 적용

파일 업로드 취약점

파일 업로드 기능이 존재하는 웹 애플리케이션에서 업로드하는 파일에 대한 적절한 유효성 검증(파일 타입 및 확장자등)을 하지 않으면 웹쉘 등의 업로드되어 서버에 악의적인 행위를 할 수 있는 취약점

<확인 방법>
- 게시판, 자료실 등 파일 업로드가 가능한 입력 폼에 악성 스크립트 파일(웹쉘) 업로드를 시도하여 업로드에 성공하면 취약한 것으로 판단

<취약점을 이용한 공격이 성공하기 위한 조건>
- 파일 업로드 디렉터리에 있는 웹쉘을 공격자가 외부에서 URL을 통해 호출될 수 있어야 하며 호출된 웹쉘이 서버 사이드 스크립트로 동작할 수 있도록 실행권한이 있어야 한다.

<대응 방안>
- 업로드 파일에 대한 파일타입 및 확장자 검증(화이트 리스트 정책 활용), 파일 크기 검증을 통해 악성 스크립트 파일(웹쉘) 업로드를 차단
- 업로드 파일을 저장하기 위한 전용 디렉터리를 별도 생성한 후 웹서버 설정을 통해 업로드 디렉터리에 있는 서버 사이드 스크립트 파일이 실행되지 않도록 설정하거나 직접 URL 호출을 차단하도록 설정

파일 삽입 취약점

공격자가 악성 서버 스크립트를 서버에 전달하여 해당 페이지를 통해 악성코드가 실행되도록 하는 취약점
삽입할 악성 서버 스크립트 파일 위치가 로컬 서버에 위치하면 LFI취약점, 원격지에 위치하면 RFI취약점으로 구분

<대응 방안>
- 관리자는 소스코드에 include또는 require등의 구문/함수가 존재하는지 검증
- 만약 사용자의 입력값을 통해 삽입할 파일명이 결정된다면 외부의 악의적인 파일이 로컬파일에 포함되어 실행되지 않도록 PHP 설정 파일인 php.ini 파일에서 'allow_url_fopen'을 Off로 설정
* allow_url_fopen기능은 PHP에서 include, require 등의 함수를 이용하여 외부 파일을 URL 형식으로 읽어올 수 있도록 해주는 기능 -> 보안상 취약한 설정

Apache 웹서버에서 파일 업로드 취약점 제거 설정

1) FilesMatch 지시자를 이용하여 파일 업로드 디렉터리에 위치한 .ph, .inc, .lib등의 서버 사이드 스크립트 파일을 찾은 후 Deny 설정을 통해 직접 URL 호출을 하지 못하게 설정
-> 파일 업로드 디렉터리에 악의적인 서버 사이드 스크립트가 업로드된 경우라도 외부에서 직접 URL로 호출하는 것을 차단하여 서버 사이드 스크립트가 실행되지 않도록 함

2) AddType 지시자를 이용하여 파일 업로드 디렉터리에 위치한 .php, .cgi 등 서버 사이드 스크립트 파일들의 MIME Type을 text/html로 재조정
-> 파일 업로드 디렉터리에 악의적인 서버 사이드 스크립트가 업로드된 경우라도 서버 사이드 스크립트파일들의 MIME 타입을 HTML 타입으로 재조정하여 서버에서 실행되지 않도록 한다

웹쉘(Webshell)

악의적인 목적으로 웹서버 내 임의의 명령을 실행 할 수 있는 서버(웹) 스크립트 파일
시스템 정보 획득, 소스 파일 변조, 시스템 명령 실행 및 원격제어 등의 악의적 행위를 수행
다양한 언어(asp, jsp, php등)로 만들어진다

개발 보안 관리의 기본 원칙

1) 최종 통제 메커니즘은 반드시 서버에서 수행되어야 한다

2) 중요정보 전송시 POST 메소드 및 SSL/TLS를 적용한다

3) 중요정보를 보여주는 페이지는 캐시를 no-cache로 설정한다

개발 보안취약점 유형

1) 입력 데이터 검증 및 표현
- 프로그램 입력값에 대한 검증 누락 또는 부적절한 검증, 데이터의 잘못된 형식 지정 때문에 발생할 수 있는 보안 취약점
- SQL 삽입, 코드 삽입, 경로 조작 및 자원 삽입, 크로스 사이트 스크립트, 운영체제 명령어 삽입 등이 발생할 수 있음

2) 보안 기능
- 보안 기능(인증, 권한 관리, 암호화 등)을 적절히 구현하지 않았을 때 발생할 수 있는 보안 취약점
- 적절한 인증 없는 중요기능 허용, 부적절한 인가, 중요한 자원에 대한 잘못된 구너한 설정, 취약한 암호 알고리즘 사용 등이 있음

3) 시간 및 상태
- 동시 또는 거의 동시 수행을 지원하는 병령 시스템, 하나 이상의 프로세스가 동작하는 환경에서 시간이나 상태를 부적절하게 관리하여 발생할 수 있는 보안취약점
- 경쟁조건, 종료되지 않는 반복문 또는 재귀함수등이 있음

4) 에러처리
- 에러를 처리하지 않거나 불충분하게 처리하여 에러메시지에 중요 정보(시스템 등)가 포함될 때 발생할 수 있는 보안취약점
- 오류 메시지 정보 노출, 오류 상황 대응 부재, 부적절한 예외 처리 등

5) 코드 오류
- 타입 변환 오류, 자원의 부적절한 반환 등과 같이 개발자가 범할 수 있는 코딩 오류로 말미암아 유발되는 보안취약점
- Null Pointer 역참조화, 부적절한 자원 해제, 해제된 자원 사용 등

6) 캡슐화
- 중요한 데이터 또는 기능성을 불충분하게 캡슐화하였을 때 인가되지 않은 사용자에게 데이터 누출이 가능해지는 보안 취약점
- 잘못된 세션에 의한 데이터 정보 노출, 제거되지 않고 남은 디버그 코드 등

7) API 오용
- 의도된 사용에 반하는 방법으로 API를 사용하거나 보안에 취약한 API를 사용하여 발생할 수 있는 보안 취약점
- DNS lookup에 의존한 보안 결정, 취약한 API 사용

HTTP 쿠키

<구분>
- Set-Cookie : 서버에서 쿠키를 생성하여 웹 브라우저에 전달할 때 사용하는 HTTP 응답 헤더
- Cookie : 웹 브라우저가 쿠키를 서버로 전달할 때 사용하는 HTTP 요청 헤더

<배포시 보안 옵션>
- Secure 옵션 : 클라이언트에서 HTTPS 통신(암호화 통신)일 경우에만 해당 쿠기를 전송하고 HTTP 통신(평문 통신)일 경우에는 전송하지 않도록 설정하는 옵션으로 쿠키 정보의 기밀성을 보장
- HttpOnly 옵션 : 클라이언트에서 스크립트를 통한 쿠키 접근을 차단하도록 설정하는 옵션으로 악의적인 쿠키 정보 탈취를 방지할 수 있다
- Expires 옵션 : 쿠키의 유효기간(만료 기간)을 설정하는 옵션으로 유효기간(만료 기간)을 최소한으로 설정하여 쿠키 정보가 클라이언트의 하드디스크 등에 지속해서 남아 공격자에게 노출되는 위험을 방지할 수 있다.

홈페이지 보안 취약점 점검 항목 - 불완전한 암호화 저장

1) 점검 방법
- DB에 저장된 중요정보가 SQL Query로 열람이 가능한지 확인
- 세션 ID나 암호화된 쿠키값이 명백히 랜덤하게 생성되는지 확인
- 적절한 암호화 알고리즘이 제대로 적용되었는지 검증

2) 조치 사항
- DB에 저장할 중요한 정보는 암호화하여 저장
- 암호화 관련 기능이나 코드 적절성의 구현 가능성을 검증한 후 사용
- 암호화가 요구되는 정보는 가능하다면 저장하지 않는다
- 기밀 정보는 최소한 두 군데 이상 분산 저장하고 실행할 때 이를 합쳐서 사용한다
- AES, RSA, SHA-256등 인정받는 알고리즘 사용

SET의 암복호화 절차

1) 암호화
- 송신자는 메시지를 압축하고, 압축된 메시지를 다시 송신자의 비밀키로 암호화하여 전자서명을 만듬
- 원문 메시지 | 에 전자서명을 첨부한 다음, 이를 대칭키로 암호화한다
- 그 다음 사용된 대칭키를 다시 수신자의 RSA 공개키로 암호화한 후 암호문과 함계 전송(전자봉투)

2) 복호화
- 전자봉투에서 수신자의 RSA 비밀키로 암호를 복호화하여 대칭키를 알아낸다
- 그후 수신자는 대칭키로 암호화된 메시지를 복호화하여, 메시지는 원문 메시지와 전자서명을 구한다
- 전자서명을 송신자의 공개키로 복호화하여, 메시지 다이제스트 값을 구한다
- 수신자는 원문메시지를 압축하여 메시지 다이제스트를 구한다

용어

hosts 파일(/etc/hosts) : IP 주소와 도메인명에 대한 매핑 정보를 담고 있는 설정 파일로 네임(DNS)서버에 질의하기 전에 먼저 참조되는 파일

resolv.conf 파일(/etc/resolv.conf) : 시스템 기본 네임 서버 설정 정보를 담고 있는 파일

php.in : PHP 환경의 설정 파일, display_error 값이 off로 되어있다면, PHP 실행 중 에러 정보 노출을 방지할 수 있다. / 개발 시에는 on으로 해놓아 자세한 에러 정보를 보며 디버깅 작업을 할 수 있다.

ftpusers : 유닉스/리눅스 시스템에서 FTP 사용자에 대한 접근 제한 / 즉, FTP 접속을 제한할 계정 정보를 담고 있는 설정 파일

폭탄 메일(mail Bomb) : 상대방에게 피해를 줄 목적으로 특정한 사람이나 특정한 시스템을 대상으로 수천, 수만통의 전자우편을 일시에 보내거나, 대용량의 전자우편을 지속해서 보내 결국 해당 사이트의 컴퓨터 시스템에 고장을 야기

스팸 메일 : PC통신이나 인터넷ID를 가진 사람에게 일방적 대량으로 전달되는 전자우편으로 발신자와 수신자가 아무런 관계가 없으며 정크메일이라고도 불림
: 무차별로 살포되어 시간과 비용을 낭비하게 하여 ISP와 PC 통신업체에서 필터링 정책을 시행하고 있으나 중계 스팸은 필터링을 우회하기도 한다
: 실시간 RBL로 대처하기도 한다.

xp_cmdshell : MS-SQP 데이터베이스의 확장 프로시저 / 데이터베이스에서 시스템 명령을 실행할 수 있도록함

스템 동작 및 운영에 악영향을 미칠 수 있는 보안 약점

딥링크(Deeplink) : 모바일 앱마다 개별적으로 생성한 딥링크에 대한 인증이 미흡할 경우 발생할 수 있음
: 공격자가 조작한 링크로 사용자가 접속하면 앱 내 자바스크립트 권한 부여를 통해 민감한 개인정보를 호출할 수 있고, 이를 통해 공격자 사이트에 개인정보가 전달될 수 있다.

robots.txt : 검색로봇에 대해 웹 사이트의 디렉터리 및 파일들에 대한 검색조건을 명시해놓은 설정 파일

nslookup / dig 명령어 : DNS 질의를 통해 DNS 서버로부터 여러가지 정보를 얻을 수 있는 명령어

웹 프록시
- 웹을 이요한 해킹공격에 가장 활발히 사용되는 도구
- 클라이언트가 요청한 HTTP Request, Response 정보를 중간에서 가로채 필터링을 우회하거나 서버에 전송되는 데이터를 조작해 허가되지 않는 곳에 정보를 훔쳐보거나 변경, 삽입 등을 가능하게 한다

PHP 환경에서 운영체제 명령을 실행할수 있는 함수
- passthru()
- exec()
- system()
- shell_exec()