보안서버(SSL) 소스 수정 매뉴얼

(사용자 매뉴얼)

순 서

본 사용자 매뉴얼은 정보보호진흥원에서 보급된
보안서버 설치 가이드를 기준으로 작성되었습니다.

□ 웹 페이지 수정 및 적용 확인하기

1. 웹 페이지 수정 방법 및 사례

암호화 통신을 하기 위해서 보안 프로토콜을 호출하는 방법은 OS나Program 언어를 가리지 않고 모두 동일합니다. 그 이유는 암호화 통신을 하기 위해 적용하는 부분이 특정 OS나 특정 Program 언어에 의존하지 않는, 모두가 공통으로 사용하는 HTML 언어이기 때문입니다.
본 절에서는 암호화 적용 범위에 따라 웹페이지 전체 혹은 일부를 암호화하는 방법과 이용자가 선별적으로 암호화를 선택하는 방법을 소개하겠습니다.

가. 전체 페이지 암호화하기

① https 프로토콜 호출하기

https 프로토콜을 호출하여 웹페이지 전체에 적용하는 방법은 그림만으로도 곧바로 이해를 할 수 있을 정도로 아주 쉽습니다.
간단히 호출하는 프로토콜을 http://에서 https://로 수정해 주기만 하면 됩니다.
<그림 1>과 <그림 2>는 암호화 통신을 하기 위해 https 프로토콜을 호출하기 전과 호출한 후의 HTML 소스코드 예입니다.

```
if ($time3 == $time4) {
echo "
<p><a href='http://[redacted].co.kr/zboard/view.php?id=not
desc=asc&no=$no' target='_top'><font size=1 color='silv
:new::-></a></p> ";
} else {
echo "<p><a href='http://[redacted].co.kr/zboard/view.php?
adnum&desc=asc&no=$no' target='_top'><font size=1 color
}
```

<그림 1> 평문 통신을 위한 HTML 소스코드

```
if ($time3 == $time4) {
echo "
<p><a href='https://[redacted].co.kr/zboard/view.php?id=noti
&desc=asc&no=$no' target='_top'><font size=1 color='silve
::new::-></a></p> ";
} else {
echo "<p><a href='https://[redacted].co.kr/zboard/view.php?i
eadnum&desc=asc&no=$no' target='_top'><font size=1 color=
}
```

<그림 2> https 프로토콜을 호출하기 위한 HTML 소스코드

② 리다이렉션(Redirection) 설정

앞서 설명을 하였듯이, 암호화 통신을 위해서는 https 프로토콜을 직접 호출을 해줘야 합니다. 하지만, 웹페이지에 접속하는 사용자들은 일일이 https 프로토콜을 붙여서 입력을 하지 않습니다. 대부분의 경우가 www.test.co.kr 또는 test.co.kr 도메인을 웹 브라우저의 주소창에 입력하고 접속하는 경우가 대부분일 것입니다. 이 때 웹 브라우저에 그냥 도메인주소만 입력하면, 웹 브라우저는 해당 도메인 앞에 http://가 붙은 것으로 판단하고 평문 통신을 하도록 합니다.
평문 통신을 하는 경우라면 문제가 없지만, 암호화 통신을 해야 할 경우에는 https://를 직접 붙여서 입력해야 하므로 여간 불편해 하지 않습니다. 리다이렉션은 현재 접속한 도메인이나 혹은 웹페이지를 강제로 다른 주소나 다른 페이지로 변경해 줌으로써 사용자들의 불편함을 감소시켜주고 자연스럽게 암호화통신을 할 수 있도록 해주는 기능입니다.

<그림 3>은 아파치 서버에서 Redirect 지시자를 써서 http://test.co.kr 또는 http://www.test.co.kr로 들어온 사용자를 강제로 https://www.test.co.kr로 리다이렉션시켜서 암호화 통신하는 예입니다.

```
<VirtualHost test.co.kr:80>
        ServerAdmin zmnkh@test.co.kr
        ServerName test.co.kr
        ServerAlias www.test.co.kr
        DocumentRoot /home/manpage
        CustomLog logs/test.co.kr-access_log common
        Redirect / https://www.test.co.kr/
</VirtualHost>
```

<그림 3> 아파치 서버에서의 Redirection

또다른 방법으로는 OS나 Web Programming 언어의 종류에 상관없이 모두 공통적으로 사용하는 HTML tag를 이용한 방법으로써, 어떤 경우에서나 적용이 가능하기 때문에 가장 많이 이용되고 있습니다.

<그림 4>은 웹페이지의 index.html에 한 줄의 소스코드를 추가함으로써 http://URL로 접속하는 사용자들을 강제로 https://URL로 리다이렉션하는 예입니다.

```
<meta http-equiv='refresh' content='0; url=https://[redacted].co.kr/index.html' target='_top'>
```

<그림 4> HTML Tag를 이용한 Redirection

위와 같이 Meta 태그를 이용하는 경우, 1초 정도 깜빡 하는 현상이 나타나기 때문에 종종 Javascript를 이용하기도 합니다.

Meta tag를 이용한 html Redirection 방법과 동일하게, 사용자들이 익숙하게 접속하는 http://www.test.com의 index 페이지에 삽입해 두면, 사용자들이 불편하게 https://라는 프로토콜을 특별히 지정해 주지 않아도, 보안을 위해서 암호화 통신이 적용된 https:// www.test.com으로 리다이렉션해주게 됩니다.

```
<script>
var url = "https://www.test.com";
window.location.replace(url);
</script>
```

<그림 5> Javascript를 이용한 Redirection

나. 페이지별 암호화하기

페이지별 암호화는 현재 위치하고 있는 페이지에서 다른 페이지로 이동할 때, 보안을 위해서 암호화된 전송을 할 것인지 아니면 평문 전송할 것인지를 선택하여 암호화하는 것을 말합니다.
부분적인 페이지 암호화를 사용하는 이유는 암호화 적용이 필요없는 부분까지 암호화를 하여 서버의 부하를 증가시키는 것을 최대한 줄일 수 있기 때문입니다.
다음 <그림 6>은 사이트의 메뉴 부분 예입니다. 이 중 '서버관련 강좌 & TIP' 메뉴를 클릭하여 이동을 하면 https가 호출되어 서버와 클라이언트간의 통신이 암호화되어 전송되고, 'Q&A' 메뉴를 클릭하여 이동하면 http가 호출되어 서버와 클라이언트간의 통신이 평문으로 이루어지게 하는 방법을 알아보겠습니다.

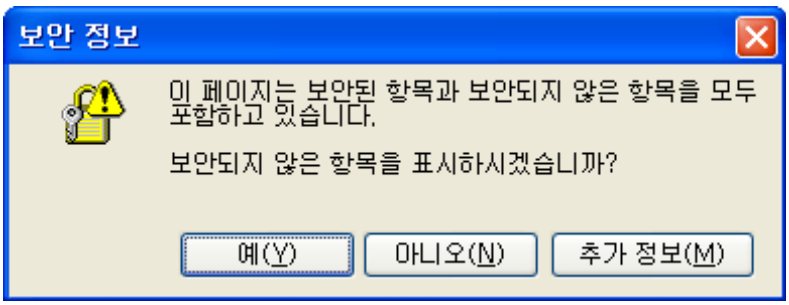
<그림 6> 페이지별 암호화 대상 메뉴

<그림 7>은 위 메뉴 부분의 소스코드입니다. 빨간색 밑줄 부분 중 첫 번째 밑줄에 해당하는 부분이 현재 위치에서 메뉴를 클릭하여 이동을 때 암호화 전송을 하도록 하게끔 설정된 것이고, 두 번째 밑줄은 현재 위치에서 메뉴를 클릭하여 이동할 때 평문 전송을 하도록 설정된 것입니다.

```
<map name="ImageMap1">
<area shape="rect" coords="193, 74, 249, 90" href="onlinebook/online.ht
m" target="main">
<area shape="rect" coords="267, 75, 401, 89" href="https://[redacted].co.k
r/zboard/zboard.php?id=lecture" target="_top">
<area shape="rect" coords="423, 73, 479, 89" href="https://[redacted].co.k
r/zboard/zboard.php?id=problem" target="_top">
<area shape="rect" coords="497, 73, 537, 89" href="http://[redacted].co.kr
/zboard/zboard.php?id=qna" target="_top">
<area shape="rect" coords="555, 73, 609, 89" href="http://[redacted].co.kr
/zboard/zboard.php?id=down" target="_top">
<area shape="rect" coords="679, 5, 717, 23" href="index.html" target="_
top">
```

<그림 7> 페이지별 암호화 대상 메뉴의 소스코드

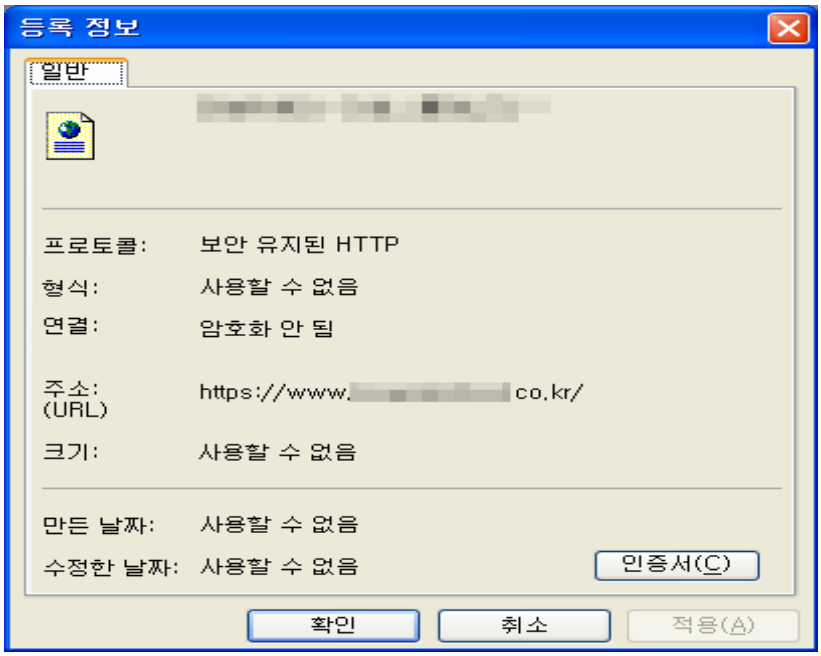
이렇게 페이지별로 암호화가 적용된 사이트를 방문해보면, <그림 8>과 같은 경고창을 만나게 되는 경우가 있습니다.



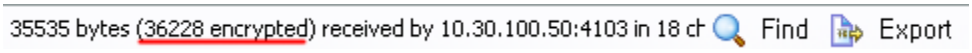
<그림 8> SSL이 적용된 페이지의 경고창

이 경고창이 뜨는 것는 암호화 통신을 유지하기 위해서는 웹페이지내의 모든 URL의 호출이 https://로 이루어져야 하나, http:// 즉 평문 통신을 위한 웹페이지 URL이 포함되어 있다는 것을 의미합니다.

이런 경고창이 발생하는 웹 페이지 속성을 보면 <그림 9>처럼 “암호화 안됨” 이라고 해서 마치 암호화가 되지 않은 평문 상태로 데이터가 전송되어지는 것처럼 생각되지만 웹 페이지간 전송되는 데이터를 볼 수 있는third-parth 툴을 이용해서 확인해 보면, <그림 10>와 같이 암호화통신이 이루어지고 있다는 것을 알 수가 있습니다.

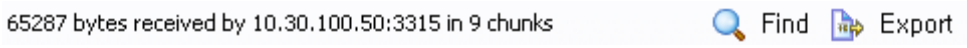


<그림 9> http 평문 통신 주소가 호출되는 웹페이지의 속성



<그림 10> https를 통한 암호화 통신

<그림 11>는 <그림 10>의 결과와 비교하기 위해서 암호화 되지 않은 평문 통신(http) 상태를 나타낸 그림입니다.



<그림 11> http를 통한 평문 통신

하지만 <그림 8>와 같이 경고창이 발생하게 되면, 상세한 내용을 모르고 웹사이트에 접속하는 사용자들에게 보안이 되고 있지 않다는 불신을 줄 수도 있고, 또한 계속적인 경고창으로 인해서 불편해 할 수 있으므로 가급적 발생하지 않도록 웹 페이지내의 모든 URL을 https://로 바꿔주는 것이 좋습니다.
만일 절대경로로 호출하는 것이 아니라, 상대경로로 호출하는 것이라면 소스를 변경하지 않아도 됩니다.

다. 프레임별 암호화하기

SSL을 이용한 보안포트(443)를 웹페이지에 적용하는 방법을 앞서 소개하였습니다. 단순히 http를 https로만 바꾸어주면 보안포트를 이용해서 암호화 통신을 할 수 있었습니다.
하지만, 프레임이 삽입된 웹페이지의 경우에는 약간 적용하는 방식이 다르기 때문에 소개하고자 합니다. 프레임이 적용된 페이지를 이용하면 암호화된 페이지와 비 암호화된 페이지를 각각 적용시킬 수 있습니다.
적용 시나리오는 <그림 12>과 같이 웹페이지(index.html)에 프레임으로 두 개의 페이지 topmenu.htm과 main.htm을 불러오는 소스코드가 있을 때 소스코드의 URL을 <그림 13>와 <그림 14>처럼 변경하고 웹 브라우저에서 http와 https로 각각 호출했을 때의 결과를 살펴보고자 합니다.

```
<html>

<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<title>SSL Frame Test</title>
</head>
<frameset rows="100, 1*" border="1">
   <frame src="http://lab.[redacted].co.kr/test_ssl/topmenu.htm" scrol
ling="yes" name="top" namo_target_frame="main">
    <frame src="http://lab.[redacted].co.kr/test_ssl/main.htm" scrolli
ng="yes" name="main">
    <noframes>
    <body bgcolor="white" text="black" link="blue" vlink="purple" ali
nk="red">
    <p>SSL Frame Test의 페이지 입니다. <br> 이페이지를 보기 위해서는
       프레임을 볼수 있는 웹 브라우저가 필요합니다.</p>
    </body>
    </noframes>
</frameset>

</html>
```

<그림 12> 프레임이 포함된 웹페이지

```
<html>

<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<title>SSL Frame Test</title>
</head>
<frameset rows="100, 1*" border="1">
   <frame src="https://lab.[redacted].co.kr/test_ssl/topmenu.htm" scro
lling="yes" name="top" namo_target_frame="main">
    <frame src="http://lab.[redacted].co.kr/test_ssl/main.htm" scrolli
ng="yes" name="main">
    <noframes>
    <body bgcolor="white" text="black" link="blue" vlink="purple" ali
nk="red">
    <p>SSL Frame Test의 페이지 입니다. <br> 이페이지를 보기 위해서는
       프레임을 볼수 있는 웹 브라우저가 필요합니다.</p>
    </body>
    </noframes>
</frameset>

</html>
```

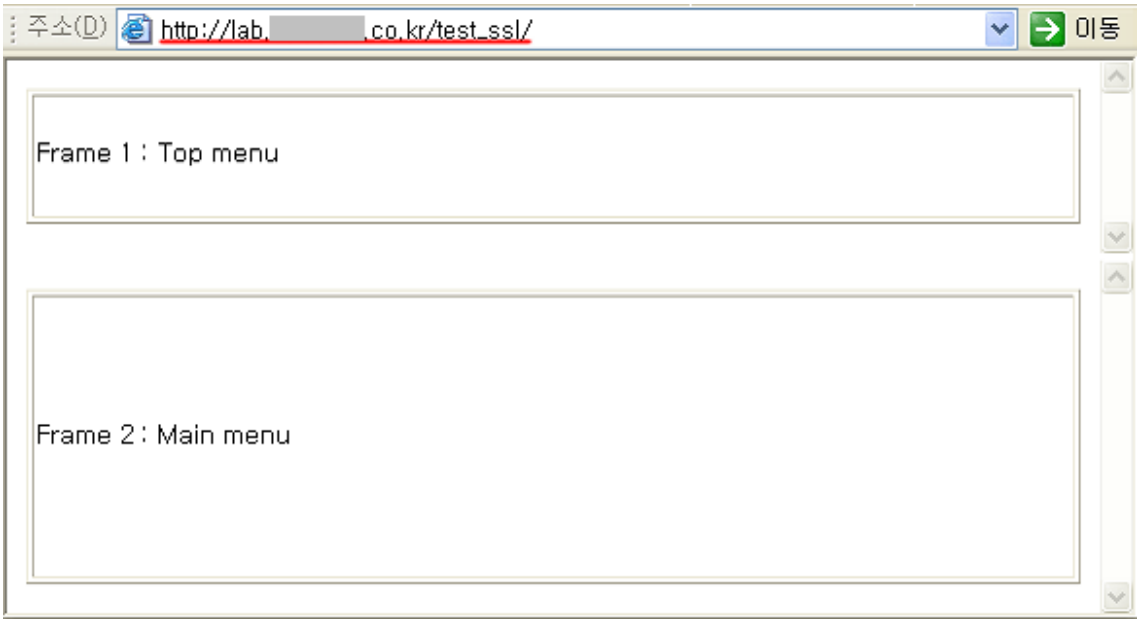
<그림 13> topmenu.htm을 https로 호출하기

```
<html>

<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<title>SSL Frame Test</title>
</head>
<frameset rows="100, 1*" border="1">
   <frame src="https://lab.[redacted].co.kr/test_ssl/topmenu.htm" scro
lling="yes" name="top" namo_target_frame="main">
    <frame src="https://lab.[redacted].co.kr/test_ssl/main.htm" scroll
ing="yes" name="main">
    <noframes>
    <body bgcolor="white" text="black" link="blue" vlink="purple" ali
nk="red">
    <p>SSL Frame Test의 페이지 입니다. <br> 이페이지를 보기 위해서는
       프레임을 볼수 있는 웹 브라우저가 필요합니다.</p>
    </body>
    </noframes>
</frameset>

</html>
```

<그림 14> topmenu.htm과 main.htm을 https로 호출하기

① 비암호화 통신(http)를 이용해서 호출하기

<그림 15>는 topmenu.htm과 main.htm을 모두 <그림 12>의 소스를 이용해서 호출한 경우입니다. 이 경우에는 모든 정보가 암호화되지 않고 <그림 16>와 같이 그대로 노출됩니다.



<그림 15> 비암호화된 페이지 호출하기

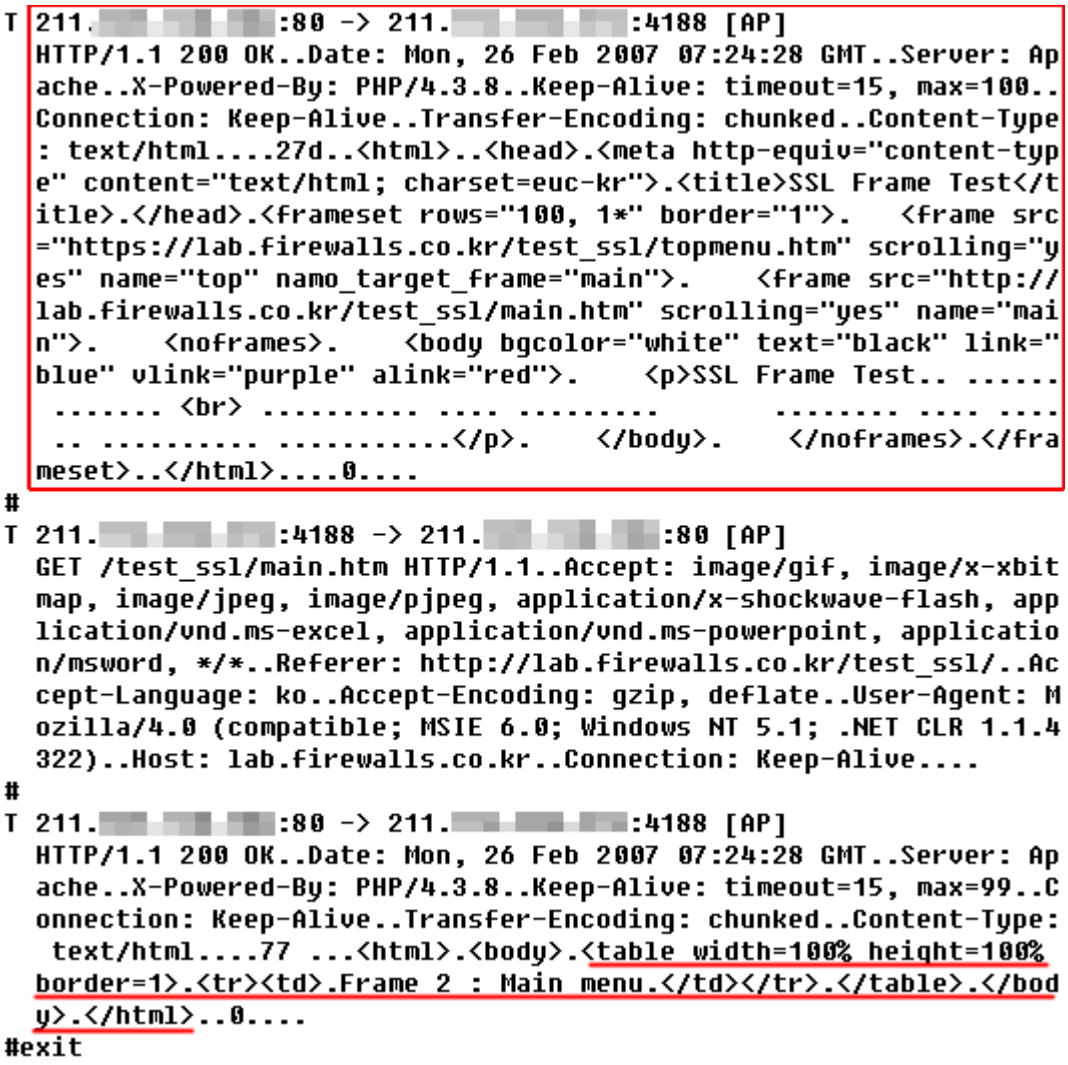
```
interface: namifw (211.[illegible]/255.255.254.0)
filter: ip and ( port 80 )
####
T 211.[illegible]:4185 -> 211.[illegible]:80 [AP]
  GET /test_ssl/ HTTP/1.1..Accept: */*..Accept-Language: ko..Accep
  t-Encoding: gzip, deflate..User-Agent: Mozilla/4.0 (compatible;
  MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)..Host: [illegible]
  [illegible]..Connection: Keep-Alive....
#
T 211.[illegible]:80 -> 211.[illegible]:4185 [AP]
  HTTP/1.1 200 OK..Date: Mon, 26 Feb 2007 07:22:59 GMT..Server: Ap
  ache..X-Powered-By: PHP/4.3.8..Keep-Alive: timeout=15, max=98..C
  onnection: Keep-Alive..Transfer-Encoding: chunked..Content-Type:
   text/html....27c..<html>..<head>.<meta http-equiv="content-type
  " content="text/html; charset=euc-kr">.<title>SSL Frame Test</ti
  tle>.</head>.<frameset rows="100, 1*" border="1">.   <frame src=
  "http://lab.firewalls.co.kr/test_ssl/topmenu.htm" scrolling="yes
  " name="top" namo target frame="main">.    <frame src="http://la
  b.firewalls.co.kr/test_ssl/main.htm" scrolling="yes" name="main"
  >.    <noframes>.    <body bgcolor="white" text="black" link="bl
  ue" vlink="purple" alink="red">.    <p>SSL Frame Test.. ..... .
  . ......... ..........</p>.    </body>.    </noframes>.</frame
  set>..</html>....0....
#
T 211.[illegible]:4186 -> 211.[illegible]:80 [AP]
  GET /test_ssl/topmenu.htm HTTP/1.1..Accept: image/gif, image/x-x
  bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
  application/vnd.ms-excel, application/vnd.ms-powerpoint, applica
  tion/msword, */*..Referer: http://lab.firewalls.co.kr/test_ssl/.
  .Accept-Language: ko..Accept-Encoding: gzip, deflate..User-Agent
  : Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.
  1.4322)..Host: [illegible]..Connection: Keep-Alive....
#
T 211.[illegible]:80 -> 211.[illegible]:4186 [AP]
  HTTP/1.1 200 OK..Date: Mon, 26 Feb 2007 07:22:59 GMT..Server: Ap
  ache..X-Powered-By: PHP/4.3.8..Keep-Alive: timeout=15, max=99..C
```

<그림 16> HTTP 호출시 80 포트 모니터링 결과

다음으로는 <그림 13>의 소스코드를 적용하여 topmenu.htm만을 https로 호출을 하는 경우입니다.

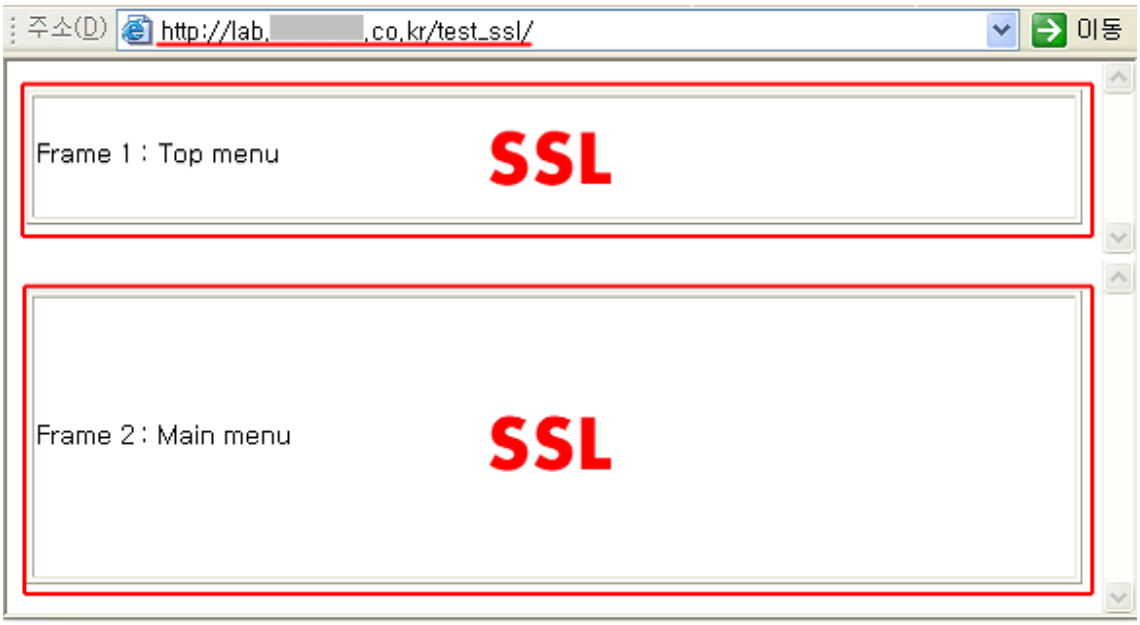
<그림 17> topmenu.htm만 암호화하여 호출하기

프레임을 이용해서 호출하는 경우에는 아래 <그림 18>과 같이 암호화되지 않는 index.html (빨간색 네모박스)의 내용과main.htm의 내용만이 80 포트로 텍스트 전송되는 것을 알 수있습니다. topmenu.htm의 내용은 암호화 전송되어지기 때문에 평문 전송되는 80 포트에서는 내용이 확인되지 않습니다.



<그림 18> topmenu.htm의 내용만 암호화된 모니터링 결과

마지막으로 <그림 14>과 같이, 호출하는 index.html을 제외하고 모든 프레임내의 호출페이지를 https를 통해서 호출하게 될 경우에는 아래와 같이 index.html의 내용만 평문으로 전송이 되고, 나머지 topmenu.htm과 main.htm은 암호화 되어서 전송됩니다.



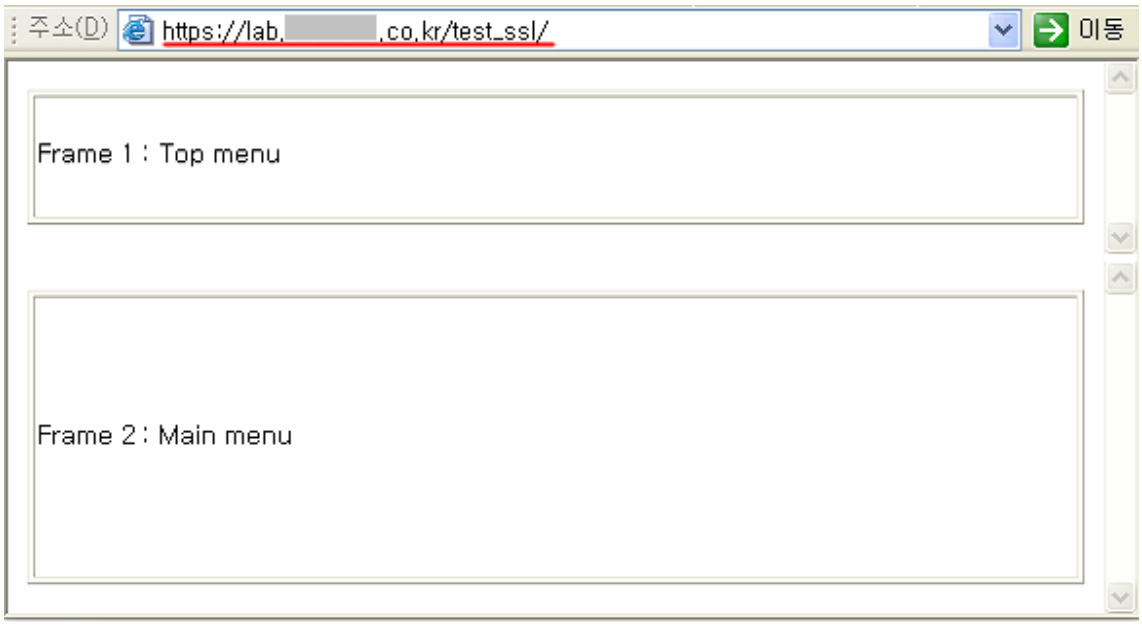
<그림 19> topmenu.htm과 main.htm을 https로 호출하기

```
interface: namifw (211.[redacted]/255.255.254.0)
filter: ip and ( port 80 )
########
T 211.[redacted]:4190 -> 211.[redacted]:80 [AP]
  GET /test_ssl/ HTTP/1.1..Accept: */*..Accept-Language: ko..Accep
  t-Encoding: gzip, deflate..User-Agent: Mozilla/4.0 (compatible;
  MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)..Host: lab.firewall
  s.co.kr..Connection: Keep-Alive....
##
T 211.[redacted]:80 -> 211.[redacted]:4190 [AP]
  HTTP/1.1 200 OK..Date: Mon, 26 Feb 2007 07:26:05 GMT..Server: Ap
  ache..X-Powered-By: PHP/4.3.8..Keep-Alive: timeout=15, max=100..
  Connection: Keep-Alive..Transfer-Encoding: chunked..Content-Type
  : text/html....27e..<html>..<head>.<meta http-equiv="content-typ
  e" content="text/html; charset=euc-kr">.<title>SSL Frame Test</t
  itle>.</head>.<frameset rows="100, 1*" border="1">.   <frame src
  ="https://lab.firewalls.co.kr/test_ssl/topmenu.htm" scrolling="y
  es" name="top" namo_target_frame="main">.    <frame src="https:/
  /lab.firewalls.co.kr/test_ssl/main.htm" scrolling="yes" name="ma
  in">.    <noframes>.    <body bgcolor="white" text="black" link=
  "blue" vlink="purple" alink="red">.    <p>SSL Frame Test.. .....
  . ....... <br> .......... .... .........       ........ .... ...
  . .. .......... ..........</p>.    </body>.    </noframes>.</fr
  ameset>..</html>....0....
####exit
```

<그림 20> index.html의 내용만이 모니터링된 결과

② 암호화 통신(https)을 이용해서 호출하기

앞에서와 같은 절차를 이용해서 https를 이용해서 <그림 11>과 같이 호출을 하게 되면, 프레임을 포함하고 있는 index.html은 URL을https로 호출을 하게 되므로, 항상 암호화가 되어지고, topmenu.htm과 main.htm은 <그림 12>,<그림 13>,<그림 14>과 같이 암호화적용 여부에 따라서, 평문 통신 또는 암호화 통신이 됩니다.



<그림 21> https를 이용한 호출

```
interface: namifw (211.[redacted]254.0)
filter: ip and ( port 80 )
#####
T 211.[redacted]:4183 -> 211.[redacted]:80 [AP]
  GET /test_ssl/topmenu.htm HTTP/1.1..Accept: image/gif, image/x-x
  bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
  application/vnd.ms-excel, application/vnd.ms-powerpoint, applica
  tion/msword, */*..Accept-Language: ko..Accept-Encoding: gzip, de
  flate..User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
   5.1; .NET CLR 1.1.4322)..Host: [redacted]..Connection:
   Keep-Alive....
##
T 211.[redacted]:80 -> 211.[redacted]:4183 [AP]
  HTTP/1.1 200 OK..Date: Mon, 26 Feb 2007 07:21:34 GMT..Server: Ap
  ache..X-Powered-By: PHP/4.3.8..Keep-Alive: timeout=15, max=100..
  Connection: Keep-Alive..Transfer-Encoding: chunked..Content-Type
  : text/html....76 ..<html>.<body>.<table width=100% height=100%
  border=1>.<tr><td>.Frame 1 : Top menu.</td></tr>.</table>.</body
  >.</html>...0....
####
T 211.[redacted]:4184 -> 211.[redacted]:80 [AP]
  GET /test_ssl/main.htm HTTP/1.1..Accept: image/gif, image/x-xbit
  map, image/jpeg, image/pjpeg, application/x-shockwave-flash, app
  lication/vnd.ms-excel, application/vnd.ms-powerpoint, applicatio
  n/msword, */*..Accept-Language: ko..Accept-Encoding: gzip, defla
  te..User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.
  1; .NET CLR 1.1.4322)..Host: [redacted]..Connection: Ke
  ep-Alive....
##
T 211.[redacted]:80 -> 211.[redacted]:4184 [AP]
  HTTP/1.1 200 OK..Date: Mon, 26 Feb 2007 07:21:35 GMT..Server: Ap
  ache..X-Powered-By: PHP/4.3.8..Keep-Alive: timeout=15, max=100..
  Connection: Keep-Alive..Transfer-Encoding: chunked..Content-Type
  : text/html....77 ...<html>.<body>.<table width=100% height=100%
   border=1>.<tr><td>.Frame 2 : Main menu.</td></tr>.</table>.</bo
  dy>.</html>..0....
##exit
```

<그림 22> https 호출시 80 포트 모니터링 결과

<그림 22>을 보면, 웹 브라우저에서 https를 통해서 호출한 index.html의 내용은 암호화되어서 통신이 이루어지기 때문에 80 포트를 모니터링 하였을 경우에 그 내용이 보이지 않지만,index.html 안에 있는 프레임을 통해서 http로 호출한 topmenu.htm과 main.htm은 https 통신을 통해서 index.html을 호출했지만, 평문으로 통신이 되는 것을 확인할 수 있습니다. <그림 13>, <그림 14>의 소스를 같은 방법으로 테스트 해보면, http로 호출된 웹페이지는 암호화 통신이 이루어지지 않고 있는 것을 알 수 있습니다.

이와 같이 프레임을 이용하면, 필요에 따라서 한 페이지에서 암호화가 제공되는 부분과 암호화가 제공되지 않는 부분이 공존할 수 있도록 구성할 수 있지만, 앞서서도 이미 언급했듯이 아무리 웹 브라우저에서 https를 이용해서 호출을 했어도 프레임으로 불러오는 페이지가 http 주소를 가지고 있을 경우에는 암호화가 되지 않고 정보의 노출이 발생할 수 있으므로, 프레임이 사용되는 페이지를 암호화를 위해서 https로 호출하고자 할 때에는 꼭 확인을 해보시기 바랍니다.

라. 체크박스를 이용한 선별적 암호화하기 웹페이지 전체를 암호화하지 않고 선별적으로 암호화하는 경우, 정보입력시 보안접속을 체크함으로써 프로토콜을 호출하는 방법이 있습니다. 다음은 로그인 박스에서 선별적으로 암호화된 통신을 하기 위한 HTML 소스코드의 예입니다.

* 소스 코드	* 소스 코드

<script language="JavaScript">
< !--
function checkLoginForm1() {
var f = document.forms["LoginForm1"];
//아이디 입력 검사
if(f.memberID.value==""){
alert("아이디를 입력하세요");
f.memberID.focus();
return false;
}
//비밀번호 입력 검사
if(f.memberPW.value==""){
alert("비밀번호를 입력하세요");
f.memberPW.focus();
	<script language="JavaScript">
	<!--
	function checkLoginForm2() {
	var f = document.forms["LoginForm2"];
	//아이디 입력 검사
	if(f.memberID.value==""){
	alert("아이디를 입력하세요");
	f.memberID.focus();
	return false;
	}
	//비밀번호 입력 검사
	if(f.memberPW.value==""){
	alert("비밀번호를 입력하세요");
	f.memberPW.focus();

```
return false;
}
//액션
f.action = "h
http://login.your-
domain.com/login1.html;
return true;
}
//-->
< /script>
< form name="LoginForm1" method="POST"
onSubmit="return checkLoginForm1();">
< table>
<tr>
<td>아이디</td>
<td><input type="text"
name="memberID"></td>
<td> </td>
</tr>
<tr>
<td>비밀번호</td>
<td><input type="password"
name="memberPW"></td>
<td><input type="submit" name="Submit"
value=" 로그인 "></td>
</tr>
< /table>
< /form>
```

```
return false;
}
//액션
if ( f.SSL_Login.checked ) { //보안접속
체크
판별
//보안접속을 체크했을 때의 액션
f.action =
"https://login.your-
domain.com/login1.html";
} else {
//보안접속을 체크하지 않았을 때의 액션
f.action =
"http://login.your-
domain.com/login1.html";
}
return true;
}
//-->
</script>
<form name="LoginForm2" method="POST"
onSubmit="return checkLoginForm2();">
<table>
<tr>
<td>아이디</td>
<td><input type="text"
name="memberID"></td>
<td><input type="checkbox" value=1
checkedname="SSL_Login" >보안접속</td>
</tr>
<tr>
<td>비밀번호</td>
<td><input type="password"
name="memberPW"></td>
<td><input type="submit" name="Submit"
value=" 로그인 "></td>
</tr>
</table>
```

	</form> ※ SSL을 이용한 암호화된 폼 전송을 하려면, action URL에서 'http' 대신 'https'를 적어 주면 됩니다.

<그림 23> 로그인시 보안접속 체크박스를 이용하기 위한 HTML 소스코드

2. 보안서버 적용 확인하기

앞에서 암호화 통신을 위해서 보안 프로토콜을 적용하는 방법을 알아보았습니다. 본 절에서는 앞서 적용한 보안 프로토콜이 제대로 적용이 되었는지 확인하는 방법에 대해서 알아보겠습니다.

① 패킷 캡쳐를 통한 확인

일단 제대로 암호화가 되어지고 있는지 패킷을 캡쳐하여 확인하는 방법입니다. <그림 24>는 일반적인 http를 통한 평문 통신의 경우를 캡쳐한 것입니다. 빨간 네모상자를 확인하면 header의 내용이 평문으로 보이는 것을 확인할 수 있습니다.

```
16:07:53.538160 [redacted].co.kr.47099 > [redacted].com.http: P 1:224(223)
op,timestamp 1156802405 1157562012> (DF) (ttl 64, id 33041, len 275)
0x0000   4500 0113 8111 4000 4006 e305 d3ef 9715        E.....@.@.......
0x0010   d3ef 96d9 b7fb 0050 a8b7 e66b ff1b 5121        .......P...k..Q!
0x0020   8018 16d0 c34c 0000 0101 080a 44f3 6765        .....L......D.ge
0x0030   44fe fe9c 4745 5420 2f20 4854 5450 2f31        D...GET./.HTTP/1
0x0040   2e31 0d0a 5573 6572 2d41 6765 6e74 3a20        .1..User-Agent:.
0x0050   6375                                          cu
16:07:53.542551 [redacted].com.http > [redacted].co.kr.47099: P 1:509(508)
,nop,timestamp 1157562013 1156802405> (DF) (ttl 64, id 44990, len 560)
0x0000   4500 0230 afbe 4000 4006 b33b d3ef 96d9        E..0..@.@..;....
0x0010   d3ef 9715 0050 b7fb ff1b 5121 a8b7 e74a        .....P....Q!...J
0x0020   8018 1920 7e3e 0000 0101 080a 44fe fe9d        ....~>......D...
0x0030   44f3 6765 4854 5450 2f31 2e31 2032 3030        D.geHTTP/1.1.200
0x0040   204f 4b0d 0a44 6174 653a 2057 6564 2c20        .OK..Date:.Wed,.
0x0050   3037                                          07
```

<그림 24> 평문 통신

```
16:13:08.640496 [redacted].co.kr.47126 > [redacted].com.https: P [tcp sum ok] 568:597(
14 win 14480 <nop,nop,timestamp 1156833916 1157593519> (DF) (ttl 64, id 22543, len 81)
0x0000   4500 0051 580f 4000 4006 0cca d3ef 9715        E..QX.@.@.......
0x0010   d3ef 96d9 b816 01bb bc26 bcaf 1261 a3a3        .........&...a..
0x0020   8018 3890 538c 0000 0101 080a 44f3 e27c        ..8.S.......D..|
0x0030   44ff 79af 1503 0100 18b1 ce57 f7b6 cce9        D.y........W....
0x0040   da65 2aba c62d eb52 d397 5bad c741 730d        .e*..-.R..[..As.
0x0050   64                                            d
16:13:08.640826 [redacted].com.https > [redacted].co.kr.47126: P [tcp sum ok] 2414:244
597 win 7504 <nop,nop,timestamp 1157593519 1156833916> (DF) (ttl 64, id 58266, len 81)
0x0000   4500 0051 e39a 4000 4006 813e d3ef 96d9        E..Q..@.@..>....
0x0010   d3ef 9715 01bb b816 1261 a3a3 bc26 bccc        .........a...&..
0x0020   8018 1d50 c670 0000 0101 080a 44ff 79af        ...P.p......D.y.
0x0030   44f3 e27c 1503 0100 18f8 0da4 2d0d 6837        D..|........-.h7
0x0040   fd55 26bc 86cc e159 1d2b 5652 a1cd c5b8        .U&....Y.+VR....
0x0050   b5                                            .
```

<그림 25> 암호화된 통신

② 웹페이지에서 확인

직접 패킷을 캡쳐해서 확인하는 방법 외에도, 웹페이지에서 간단히 암호화가 되어지고 있는지를 확인하는 방법이 있습니다.
SSL이 적용된 웹사이트에 https 프로토콜로 접속을 했을 경우에, <그림 26>과 같이 브라우저 하단에 자물쇠 모양의 표시가 나타나는 것을 확인할 수 있을 것입니다. 현재 사이트와의 통신이 암호화되어서 진행되고 있다는 것을 보여주는 것입니다. 웹사이트 구성방법에 따라 자물쇠 이미지가 보이지 않을 수 있으며, 구축 방법에 따라 모양은 다르게 나타날 수 있습니다.

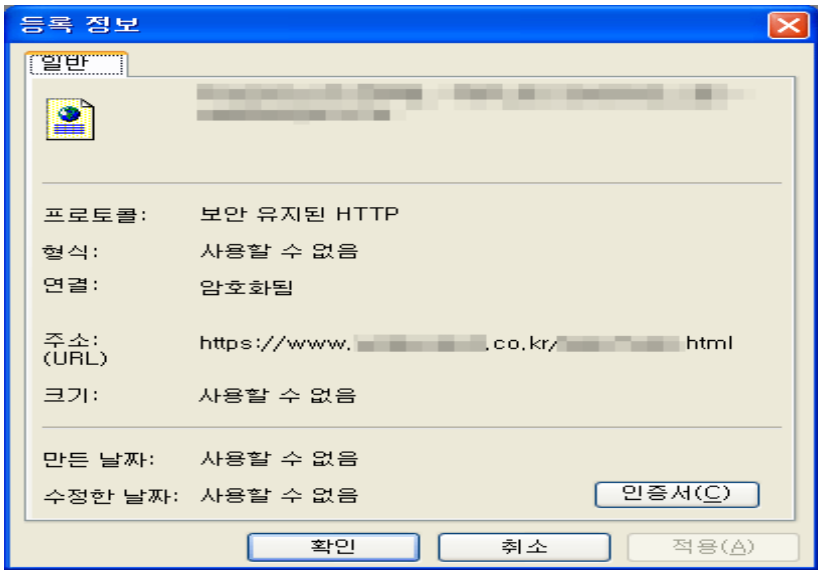
<그림 26> 암호화 통신이 이루어지고 있음을 보여주는 자물쇠 이미지

③ 호출시 포트번호 확인

https를 이용하여 접속하시면 일반적으로 443번 포트로 연결이 되어 SSL 인증서버를 통한 통신을 하게 됩니다.
서버에 여러 개의 인증서버를 설치할 경우 상황에 따라 443번 포트가 아닌 여러 가지 포트를 이용해서 접속을 하는 상황이 발생할 수 있습니다.
이런 경우 설치를 대행하는 업체나 호스팅업체에서 임의의 포트를 지정하거나 사용할 포트를 지정받아 SSL 인증서를 설치한 뒤 작업완료 통보와 함께 사용된 포트번호를 알려주게 됩니다.

④ 웹페이지 속성보기를 통한 확인

암호화를 적용한 웹페이지가 정상적으로 암호화되고 있는지는 웹페이지에서 오른쪽 마우스를 클릭하고 속성을 선택한 후 웹페이지 등록 정보를 통하여 확인할 수가 있습니다. 현재 페이지가 보안이 되고 있다면 <그림 27>와 같은 웹페이지 속성을 확인할 수 있습니다.



<그림 27> 보안이 적용된 웹페이지 등록정보