RE:Real_Name
&
RE:Anonymous

송준혁(U_SoRang)

2012.05.27. 작성

Index

1. 개요

☞ 싸이월드 클럽 게시판의 Source code를 이용함으로써 익/실명 댓글을 마음대로 조작 할 수 있다.

2. 목적

☞ 클럽 게시판 Source code의 취약성 확인과 악용 가능성 검증.

3. 과정

A. 문제인식

☞ 각 web site의 source code를 보고 contents가 어떠한 형식으로 배치되어 있는지, 어떠한 web language가 쓰였는지 알아보는 재미를 즐기는 나는 어느 날, 싸이월드 클럽 게시판의 source code를 보다 주석 처리가 되어 있는 부분을 발견했는데, 그 부분의 전송 방식은 "get"이며, 그것의 특징인 각종 parameter들이 들어가 있다는 것을 알게 되었다. "post" 방식보다 비교적 보안성이 낮은 "get" 방식을 사용함으로써 어떠한 보안상의 문제를 야기 시킬 수 있다는 생각이 들었다.

B. 가설설정

☞ 주석 처리된 "get" 방식의 특정 부분이 게시판의 한 부분을 뜻한다면, 이 주소를 이용하여 댓글 등의 여러 조작을 가할 수 있을 것이다.

C. 준비물

☞ Web browser, 실험대상인 싸이월드 클럽 게시판
〈실험환경〉 : Windows 7 Ultimate, Google Chrome 19.0.1084.52 m

D. 실험방법 및 순서

▶ 게시판에 글과 댓글을 써 보았다.
[익명 게시판]이기에 글과 댓글의 작성자가 "익 명" 으로 표시된다.

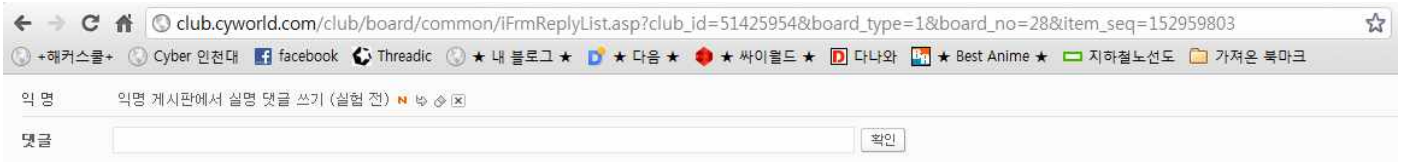
```
--> <a href="#" onClick="javascript:DeleteItem('0', '0', '0'); return false;">삭제</a> <i>|</i> <a href="#" onClick="javascript
false;"><b>보내기</b></a> <i>|</i>  <a href="#" onClick="ScrapBook(); return false;"><b>스크랩</b></a> <i>|</i>  <a href='reply
club_id=51425954&board_no=28&item_seq=152959803&search_type=&search_keyword=&cpage=1&list_type=2&board_type=1&show_type=1'><b>답글</b></a> <i>|</i>  <a h
onclick="OpenPrintView(152959803);return false;"><b>인쇄</b></a>
                        </span>

                <!-- 보내기 레이어 -->
                        <div class="lyr_common lyr_sendContent">
                                <div class="lyr_wrap">
                                <a href="#" title="판(광장)보내기" class="hideLyr" onclick="javascript:SendPann('1','28','152959803','0');return false;">
                                </div>
                        </div>

                <!-- 광장내보내기 -->
                </div><div class="clear"></div>

                <!-- iframe  class="reply_iframe" src="/club/board/common/iFrmReplyList.asp?club_id=51425954&board_type=1&board_no=28&item_seq=152959803"
id="reply_152959803" title="게시판 댓글 목록" noresize="noresize" scrolling="auto" frameborder="0"></iframe -->
```

인쇄

▶ 댓글 부분에서 마우스 팝업 메뉴를 호출 후, [프레임 소스 보기]를 선택한다.
댓글 전 부분에 있는 "인쇄" 로 검색하면, 주석 처리된 부분을 볼 수 있다.

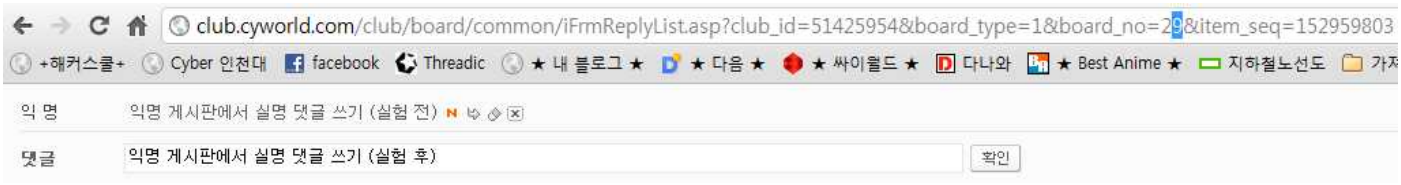
```
                <!-- iframe  class="reply_iframe" src="/club/board/common/iFrmReplyList.asp?club_id=51425954&board_type=1&board_no=28&item_seq=152959803"
id="reply_152959803" title="게시판 댓글 목록" noresize="noresize" scrolling="auto" frameborder="0"></iframe -->
```

▶ 주석 부분에서 src 뒤의 문자열이 이 익명 게시판을 뜻하는 주소일 것이다. 확인하기 위해 access를 하겠다.
("/club/board/common/iFrmReplyList.asp?club_id=51425954&board_type=1&board_no=28&item_seq=152959803")
하지만, 그대로 aceess할 수 없기에, 앞에 "http://club.cyworld.com" 을 붙이도록 한다.

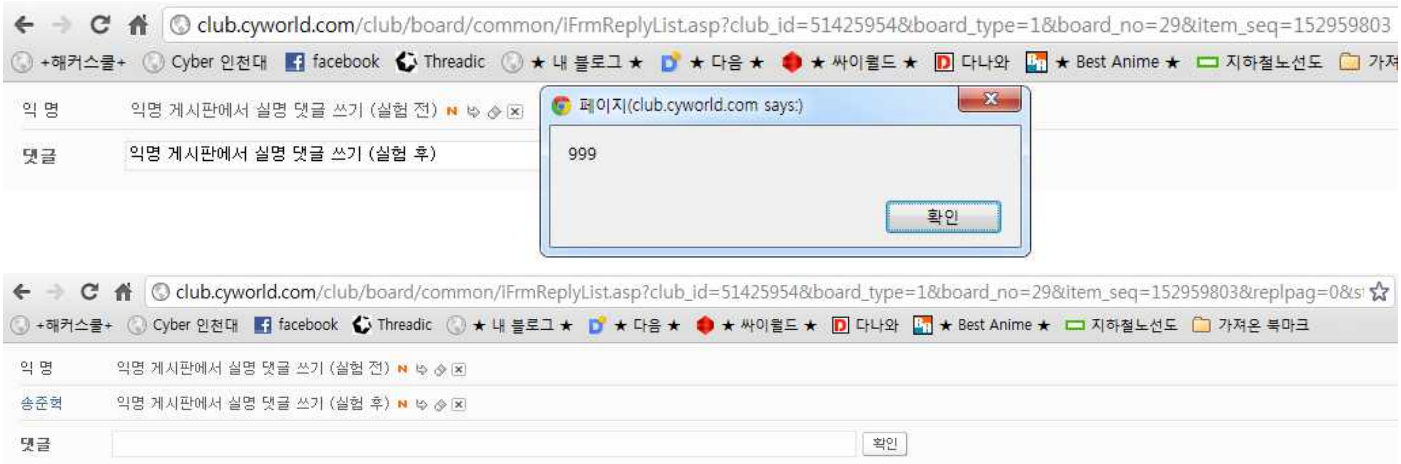


▶ 들어가면 위의 그림과 같이, 댓글 부분만 불러올 수 있게 된다.
또한, 익명 댓글이 달려 있으므로 이는 익명 게시판의 댓글 부분이 확실하다.
따라서, board_no의 값은 각각의 게시판을 구분 짓는 기능을 한다는 것을 알 수 있다.
하지만 여기에 익명이 아닌 댓글을 달기 위해서는, 실명 게시판의 board_no가 필요하다.
마찬가지 방법으로 실명 게시판의 board_no를 알아본다.

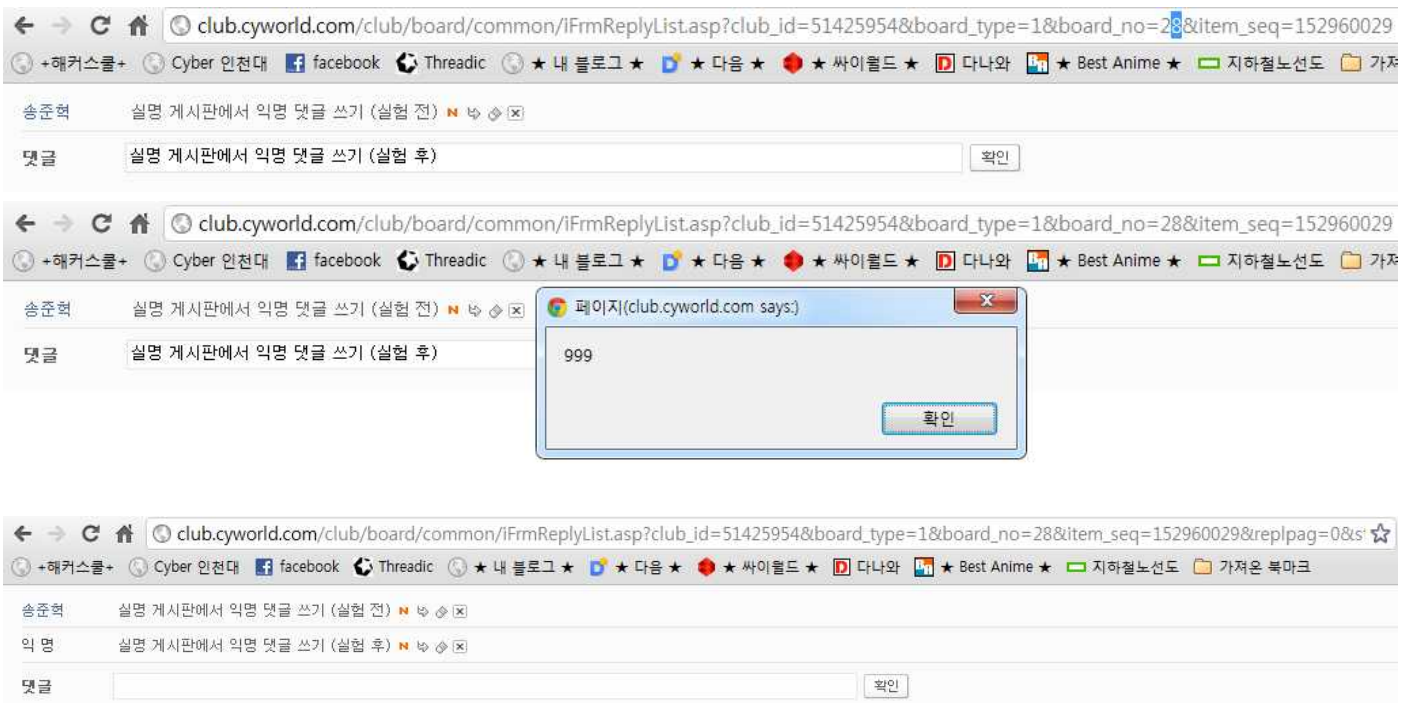
▶ 실명 게시판의 board_no는 "29" 이다.
이로써, 익명 게시판에서 실명 댓글을 쓸 수 있는 조건이 갖춰졌다.



▶ board_no가 "28" 이었던 익명 게시판을 "29" 로 수정하고, 댓글을 작성한다.



▶ 에러 메시지 출력 후, 실명 댓글이 달려 있는 것을 확인 할 수 있다.
반대의 경우도 가능하다.



▶ 반대의 경우인 <실명 게시판에서 익명 댓글 쓰기>의 결과이다.

E. 실험결과
☞ 익/실명 댓글 조작이 성공하였다.

F. 실험결론
☞ 게시판 page의 source code에 있는 주석 부분과 parameter를 이용하여 익/실명 댓글의 조작이 가능하다.

G. 가설검증
☞ 주석 처리된 특정 부분은 게시판의 한 부분을 뜻하였고, 이 주소를 이용하여 댓글 등의 여러 조작을 가할 수 있다.

4. 요구조건 및 애로사항

☞ 처음에는 익명과 실명을 나누는 code value가 있다고 생각해, 초기 실험 대상(위에서 언급하지 않은 별도의 클럽 게시판)에서 추출한 "214 / 244" 란 값을 code value로 정의하였다. 하지만, 여러 parameter를 조작하는 실험을 거듭할수록 초기의 결론이 틀렸음을 깨달았다.

초기 실험결과를 폐기하고, 새로운 실험을 시작하였다. 가설을 재설정하고 본 실험을 뒷받침해주는 선행 실험(get 방식으로 전달된 parameter값의 분석)을 먼저 진행하였다. 또 다른 특정 code value가 있을 것이란 생각에 각각의 parameter마다 임의의 값을 집어넣고 결과를 살펴보는, 그야말로 "삽질"이라 불리는 것을 반복한 결과, 익/실명 구분 code는 따로 존재하지 않으며 "게시판 생성 번호"를 뜻하는 board_no의 값이 조작을 가능케 하는 열쇠라는 것을 알게 되었다.

이 사실에 의해 익/실명 댓글 조작이 성립하기 위해서는 **["익명 게시판"이 반드시 존재 할 것.]** 과 **[익명과 실명 게시판의 "board_no" 값을 알아야 한다.]**

5. 결론

☞ 이 실험에 의해, "get" 전송 방식을 쓰는 싸이월드 클럽 게시판은 주석 부분을 이용하여 얼마든지 댓글 조작을 가할 수 있다. 아직 모든 parameter들에 대한 정확한 정보는 얻어내지 못했지만, 이 또한 밝혀진다면 어떤 형식으로든 게시판을 사용자 마음대로 요리할 수 있을 것이다.

"get" 전송 방식은 전송되는 parameter들이 쉽게 노출됨으로써 위 실험 결과 같은 보안상의 문제점을 야기 시키기 때문에, 전송 시 header에 parameter를 숨겨서 전송하는 "post" 방식에 비해서 보안성이 낮다.

따라서 현 전송 방식을 변경한다면, 비교적 보안상의 문제가 줄어들지 않을까라고 생각해본다.