Broken Authentication and Session Management Basic

윤현호 (Hyunho.Yun)

r3dcat@gmail.com

Published:September 2007

http://theFlower.or.kr

Abstract

- 본 문서는 2007년 9월 7일 SecurityPlus(café.naver.com/securityplus) OWASP Top10 2007 Seminar 관련 자료입니다.
- 영리를 목적으로 사용 및 배포를 금지합니다.
- 문서의 내용은 임의의 가상 테스트 서버를 대상으로 한 기본적인 OWASP TOP 10 범위 안에서의 기본적인 내용들로 구성되어 있습니다.
- 더욱 자세한 내용은 http://gimyo.com/owasp 를 참고해주시기 바랍니다.
- 본 문서는 OWASP의 이해를 돕기 위한 문서이며, 비인가 된 접근은 불법입니다.

1 Introduction

이번 세션에서 다룰 이야기는 취약한 인증 및 세션관리 입니다.

우선 웹App의 인증처리 방법들에 종류와 실제로 일어나는 자바스크립을 통한 인증확인 모듈을 쉽게 우회하는 방법 그리고 마지막으로 플래쉬 이벤트 게임의 결과를 조작하여 이벤트 상품을 취득하는 방법에 대해서 알아보겠습니다.

2 상세내용

아시는 바와 같이 HTTP는 비연결지속형(stateless) 프로토콜이죠. 그렇기 때문에 어떤 사이트에 방문을 해도 서버는 해당 사용자가 처음으로 방문한 것으로 인지합니다. 그렇지만! 실제로 우리는 로그인을 하고 각 페이지들을 서핑하는데 아무런 문제가 없습니다. 바로 세션관리모듈을 통해 우리는 아무런 불편 없이 서핑을 할 수 있는 것 입니다. 그래서 쿠키나 세션을 통해 사용자를 구분하기도 하고 히든 값을 통해서 어떤 정보를 각 페이지에 넘겨주고 받기도 합니다. 이런 구조적인 문제로 인해 쿠키 값 위변조, 세션 위변조, 인증우회 등의 문제들이 발생하고 있습니다.

가) JavaScript로 된 주민등록번호 확인 모듈 우회

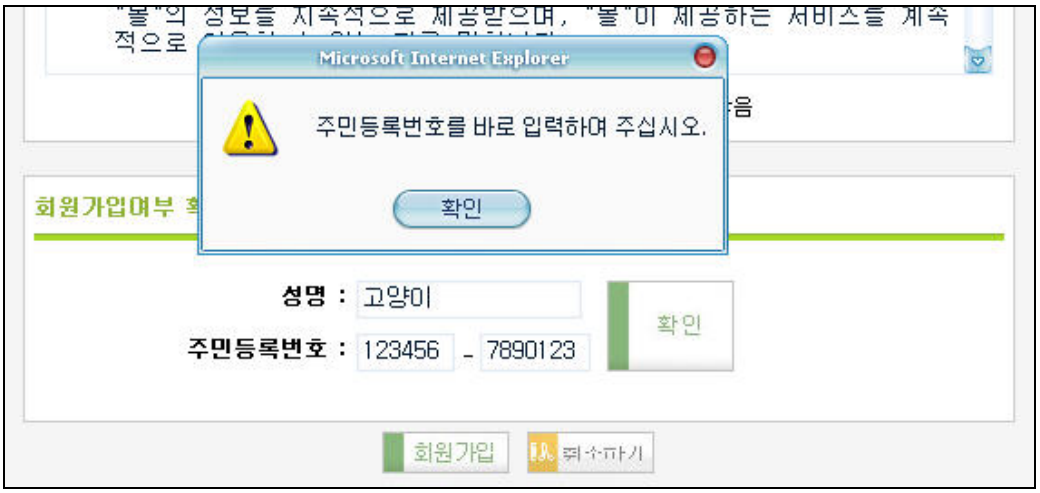
이번에 살펴볼 것은 자바스크립 만으로 인증을 하는 것이 왜 문제인지 살펴보겠습니다. 해당 사이트는 주민등록번호 인증 모듈을 갖추고 있어 적법한 형식의 주민등록 번호 외에는 가입이 되지 않습니다. 이 모듈을 쉽게 우회하는 방법을 보도록 하겠습니다.

.

위의 그림과 같이 회원가입 여부를 확인하기 위해서 주민등록번호를 입력 받는 사이트들이 많습니다. 이때 주민등록번호가 그 형식에 맞는지를 확인하여 위조된 주민등록번호를 입력 받지 않도록 하는 모듈을 함께 구현 합니다.

이때 인증 관련 모듈이 JavaScript로 작성되었을 경우 이를 쉽게 우회할 수 있습니다.

실습용 사이트에 가입 신청을 해보겠습니다. 주민등록번호 형식에 맞지 않는 번호를 넣어보겠습니다.



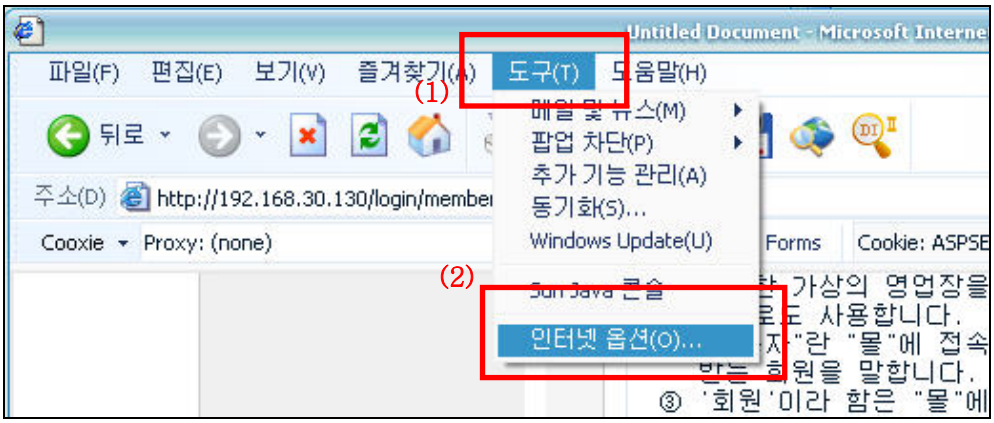
올바른 주민등록번호를 입력 하라는 메시지 창이 뜨는 것을 볼 수 있습니다. 그렇지만 '확인' 버튼을 누르는 사이에 사이트가 재로딩이 되지는 않고 있습니다. 자바스크립트를 이용해서 입력값을 검증할 가능성이 높기 때문에 소스보기를 해보도록 하겠습니다.

.

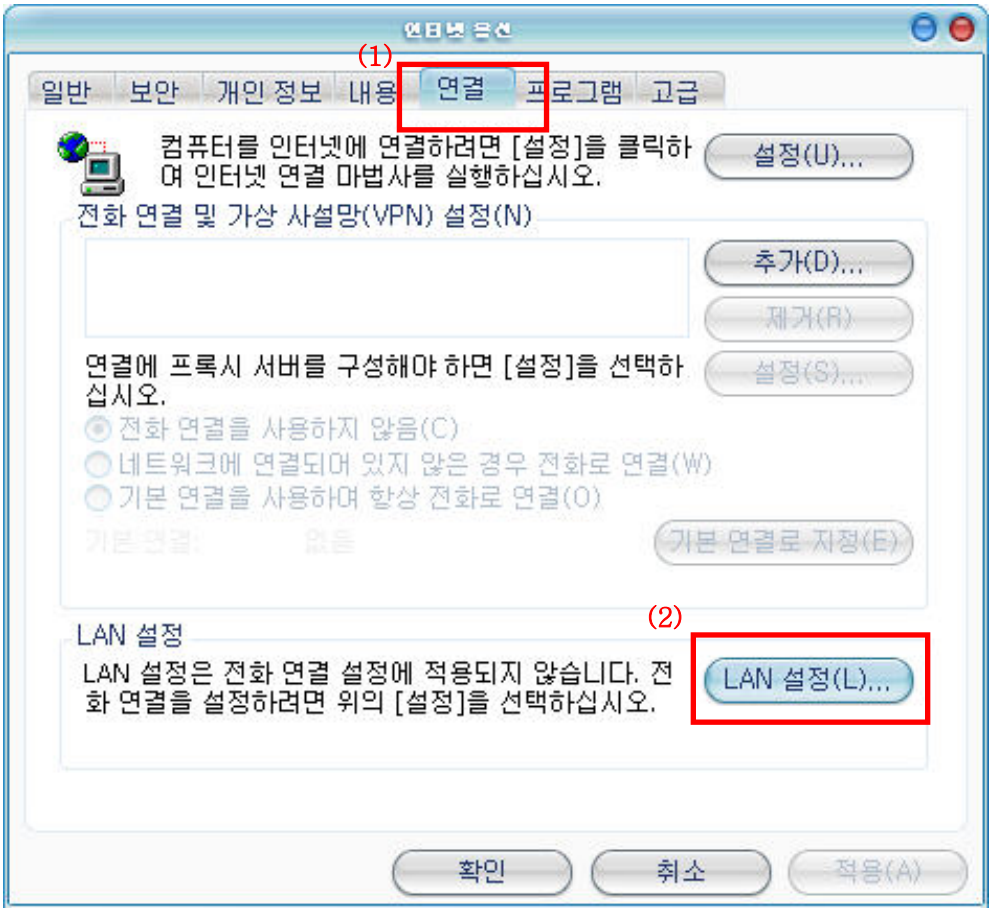
```
82              if ((chkSex != 1 && chkSex !=2 && chkSex !=3 && chkSex !=4) || (document.for
83              {
84                      alert ("주민등록번호를 바로 입력하여 주십시오.");
85                      document.form.mem_jumin2.focus();
86                      return;
87              }
88
89              // 주민등록번호 validation check
90
91              for (var i = 0; i <=5 ; i++)
92              {
93                      chk = chk + ((i%8+2) * parseInt(document.form.mem_jumin1.value.subst
94              }
95              for (var i = 6; i <=11 ; i++){
96                      chk = chk + ((i%8+2) * parseInt(document.form.mem_jumin2.value.subst
97              }
98              chk = 11 - (chk %11)
99              chk = chk % 10
```

소스를 통해서 확인 결과 JavaScript으로 '주민등록번호를 바로 입력하여 주십시오.'라고 메시지 창을 띄운 소스가 나타납니다. 그리고 나머지 소스도 살짝 살펴본 경과 주민등록번호 확인 모듈이 모두 JavaScript로 구현된 것을 알 수 있습니다.

이번에는 로컬 웹 Proxy인 Paros를 사용해 보도록 하겠습니다.



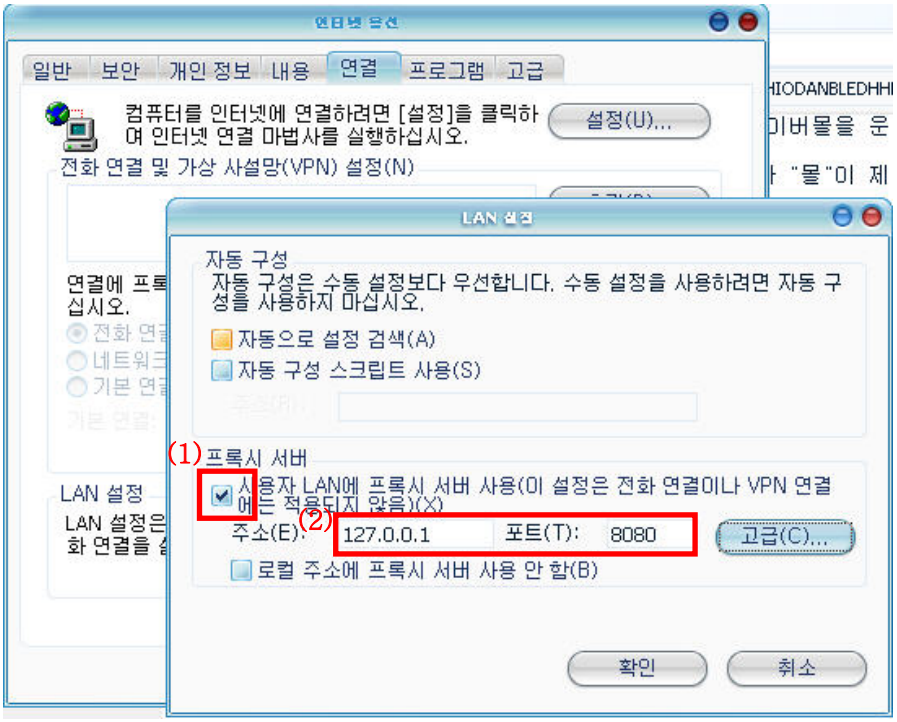
IE의 메뉴바에서 '도구(T)'를 선택 '인터넷 옵셥(O)'을 선택 합니다.



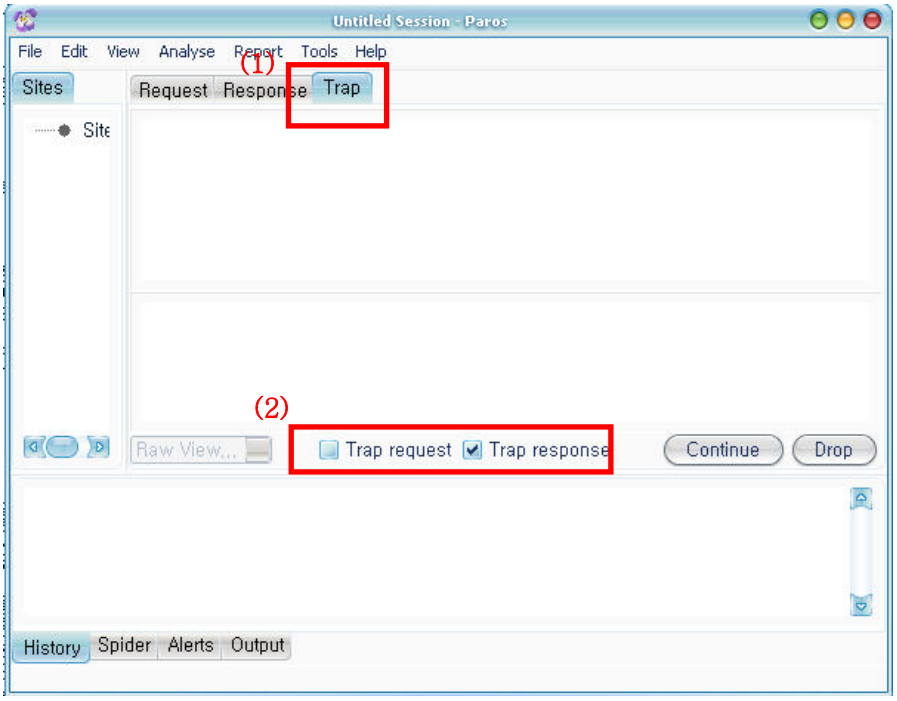
'연결' 탭의 'LAN설정(L)'을 클릭 합니다.

그리고 '프록시 서버' 에서 체크 항목에 체크를 해주신 후 '주소(F)'란에 127.0.0.1 포트 8080 으로 입력을 해줍니다.

이렇게 입력을 하면 127.0.0.1은 로컬IP를 뜻하기 때문에 IE에서 접속을 할 경우 모두 로컬 포트인 8080포트를 통해서 연결을 하도록 설정 하는 것 입니다.

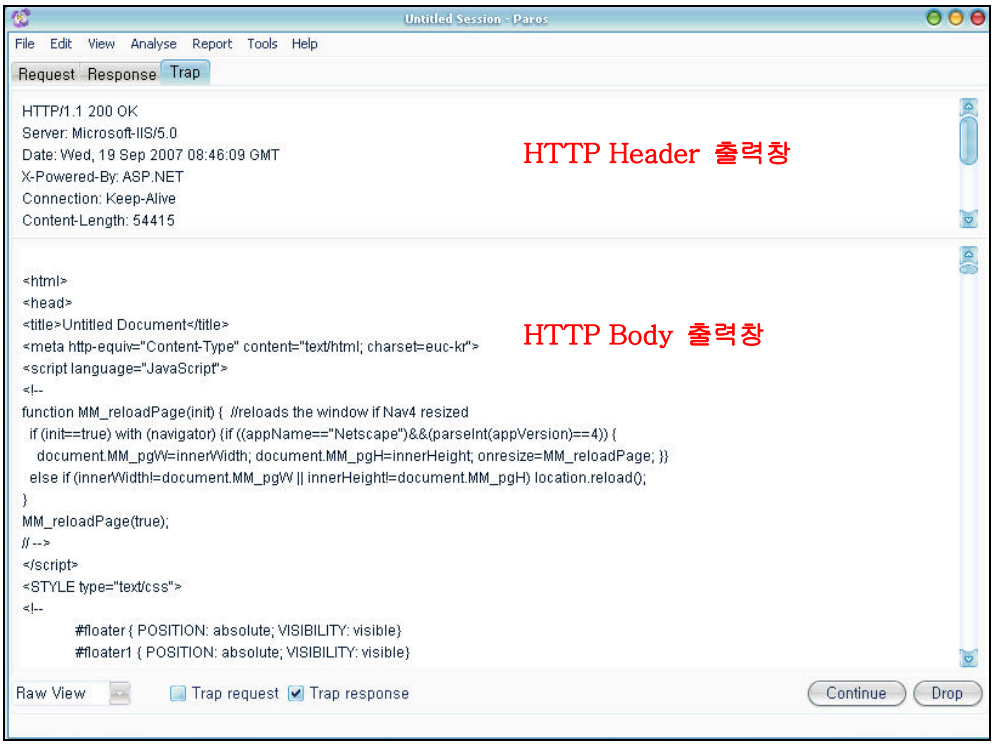


이렇게 설정을 마치셨다면 미리 인스톨 해둔 Paros를 작동 시킵니다.



'Trap' 탭을 선택 한 후 'Trap response' 항목을 체크 합니다.

그리고 준비가 다 되었습니다. IE의 새로고침 페이지를 눌러주세요. (F5키)
그러면 다음과 같은 결과가 돌아오는 것을 확인 할 수 있습니다.



그러면 돌아온 결과에서 HTTP Body 출력창에서 IE에 전달하기 전의 데이터 값을 가로챈 것을 확인 할 수 있습니다.

IE에 전달되기 전의 JavaScript 부분을 제거하여 입력 값 검증을 우회하도록 하겠습니다.

우선 아래 천천히 바를 내려서 살펴보면 /////주민등록 중복 여부///라는 주석을 쉽게 발견 할 수 있습니다.

그 아래에 보면 번호를 체크하는 모듈이 있음을 쉽게 알 수 있는데 바로 그 부분을 다 제거 하도록 하겠습니다.

```
if ((document.form.mem_jumin1.value.length != 6 ) || (mm < 1 || mm > 12 || dd < 1 || dd > 31 ) )
{
alert ("주민등록번호를 바로 입력하여 주십시오.");
document.form.mem_jumin1.focus();
return;
```

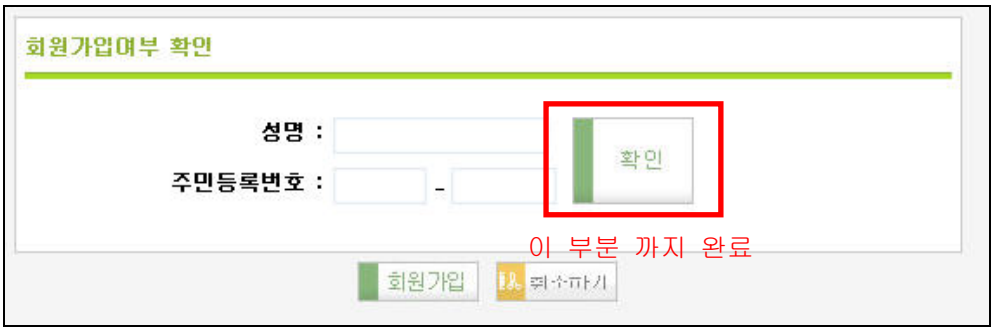
```
        }

        if ((chkSex != 1 && chkSex !=2 && chkSex !=3 && chkSex !=4) ||
(document.form.mem_jumin2.value.length != 7 ))
        {
                alert ("주민등록번호를 바로 입력하여 주십시오.");
                document.form.mem_jumin2.focus();
                return;
        }
```

위의 파란색으로 표기한 부분이 첫번째 확인 부분입니다. 이 부분을 모두 삭제하도록 하겠습니다. 해당 부분을 드래그해서 Delete키로 삭제 합니다.
그 아래로 좀더 내려가면 다시 두번째 체크 구문이 나타납니다. 이 부분도 모두 제거를 해줍니다.

```
if (chk != document.form.mem_jumin2.value.substring(6,7))
        {
                alert ("유효하지 않은 주민등록번호입니다.");
                document.form.mem_jumin1.value=""
                document.form.mem_jumin2.value=""
                document.form.mem_jumin1.focus();
                return;
        }
```

여기까지가 '중복확인' 부분에서의 주민등록번호 확인 구문입니다.'



이번에는 그 아래 있는 '회원가입' 부분에 적용된 JavaScript를 제거하도록 하겠습니다. 계속 바를 아래로 내려봅니다.
////회원동의/// 밑으로 좀더 내려보면 다시 주민등록번호 체크 구문이 있습니다.

위에서와 같이 해당 부분을 제거 합니다.

```
if ((document.form.mem_jumin1.value.length != 6 ) || (mm < 1 || mm > 12 || dd < 1 || dd > 31 ) )
        {
                alert ("주민등록번호를 바로 입력하여 주십시오.");
                document.form.mem_jumin1.focus();
                return;
        }

        if ((chkSex != 1 && chkSex !=2 && chkSex !=3 && chkSex !=4) || (document.form.mem_jumin2.value.length != 7 ))
        {
                alert ("주민등록번호를 바로 입력하여 주십시오.");
                document.form.mem_jumin2.focus();
                return;
        }
```

그리고 조금 밑에 역시 있는 유효주민등록번호 확인 부분을 제거 합니다.

```
        if (chk != document.form.mem_jumin2.value.substring(6,7))
        {
                alert ("유효하지 않은 주민등록번호입니다.");
                document.form.mem_jumin1.value=""
                document.form.mem_jumin2.value=""
                document.form.mem_jumin1.focus();
                return;
        }
```

그리고 위에서는 없었지만 회원가입 여부를 확인 하는 구문이 있는 것을 알 수 있습니다. 이 부분을 제거하면 위에서 했던 확인 작업을 거치지 않아도 됩니다. 이 부분 역시 제거 합니다.

```
        if(document.form.check_jumin.value!="yes"){
                alert("회원가입여부를 확인하시기 바랍니다.")
                return;
        }
```

그리고 다시 나타난 창에 주민등록번호를 아무거나 치고 회원가입을 클릭 합니다.

위와 같이 아무번호나 입력이 되는 것을 알 수 있습니다.

이렇듯이 간단하게 JavaScript로 된 구문을 우회할 수 있었습니다. 그리고 또한 JavaScript로 인증관련 모듈이 코딩 되어 있기 때문에 그 내부 알고리즘을 쉽게 알 수 있었습니다.

나) 이벤트 Flash Game 결과 조작

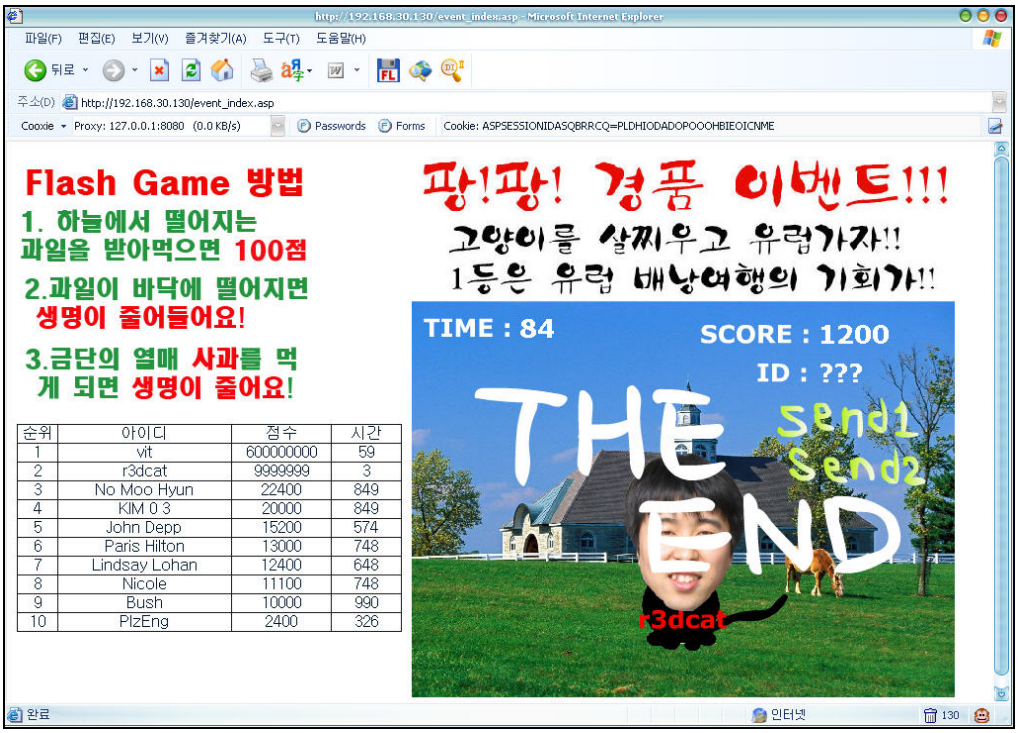
많은 다수의 사이트에서 상품의 광고 홍보 목적으로 단순 플래쉬 게임으로 상품을 나누어주는 행사를 많이 실시 합니다. 이번 세션에서는 '이벤트 게임' 결과를 조작하는 방법을 살펴보겠습니다.

우선 '이벤트 상품'를 클릭 하고 Flash Game을 Start 버튼을 클릭해서 시작하도록 합니다.

게임의 구성은 TIME항목과 SCORE항목으로 순위를 결정 합니다.

Flash 게임을 진행하는 동안에는 어떤 패킷도 외부로 전송되지 않는 것을 확인하였습니다.

우선 아래와 같이 Flash 게임 자체가 IE에서 실행이 되고 있음을 알 수 있습니다.

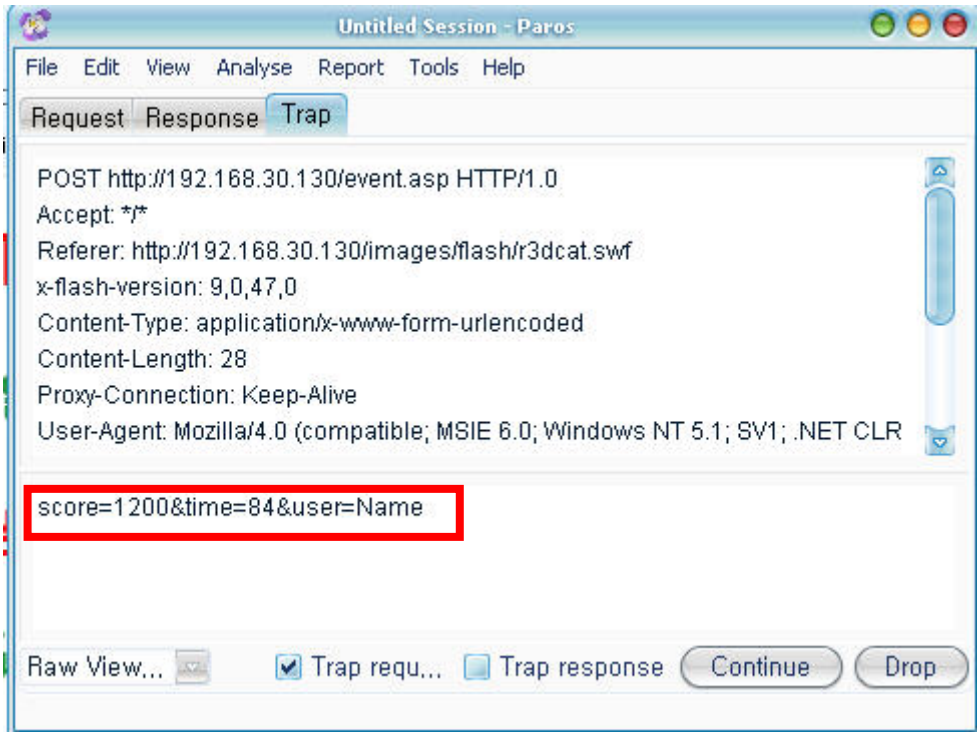


순위	아이디	점수	시간
1	vit	600000000	59
2	r3dcat	9999999	3
3	No Moo Hyun	22400	849
4	KIM 0 3	20000	849
5	John Depp	15200	574
6	Paris Hilton	13000	748
7	Lindsay Lohan	12400	648
8	Nicole	11100	748
9	Bush	10000	990
10	PlzEng	2400	326

그렇기 때문에 JavaScript 인증 모듈을 우회 했던 것과 같이 Paros를 이용해서 결과를 조작해보겠습니다. 위에서와 같은 방법으로 Paros를 다시 재설정 한 후에 실행 합니다.

이번에는 Trap request 부분을 체크 합니다.

Flash 게임에 이름을 입력하고 'Send1'을 클릭합니다. 그러면 아래 그림과 같이 score와 time 과 user 값이 전송되려고 하는 것을 알 수 있습니다.



이 부분의 값을 원하는 값으로 변조 합니다.

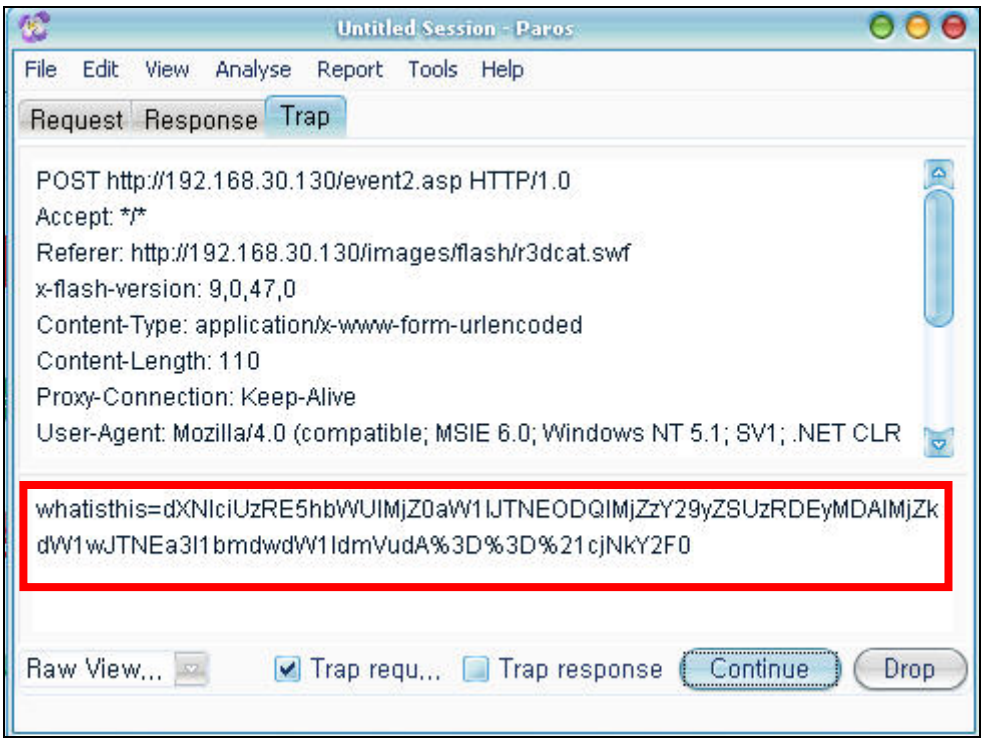
전 1등이 되기 위해서 스코어를 엄청나게 올렸습니다. 그리고 Continue 버튼을 클릭해서 전송합니다.

순위	아이디	점수	시간
1	test	900000000	84
2	vit	600000000	59
3	r3dcat	9999999	3
4	No Moo Hyun	22400	849

1위로 조작된 것을 알 수 있습니다. 어떤 세션 값 검증이 없는 경우 이렇게 허무하게 값이 삽입이 되는 것을 알 수 있습니다.

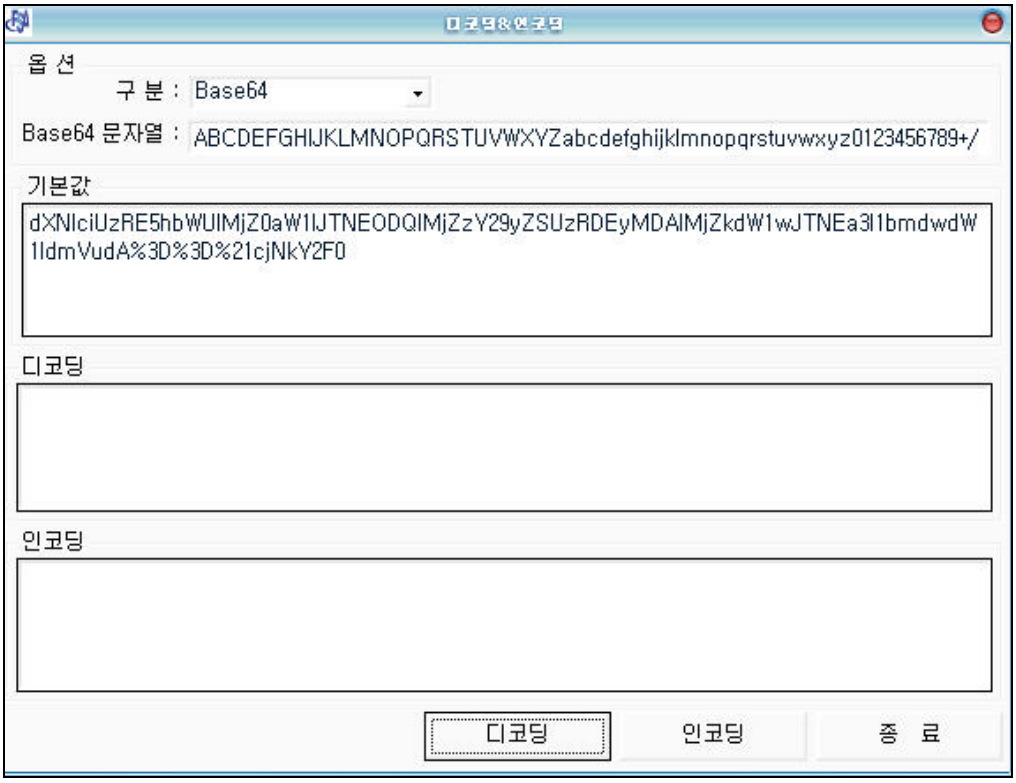
그렇지만 여러 사이트에서는 평문으로 결과를 전송하지 않는 경우도 많습니다.

이번에는 send 2를 클릭 해보도록 하겠습니다.

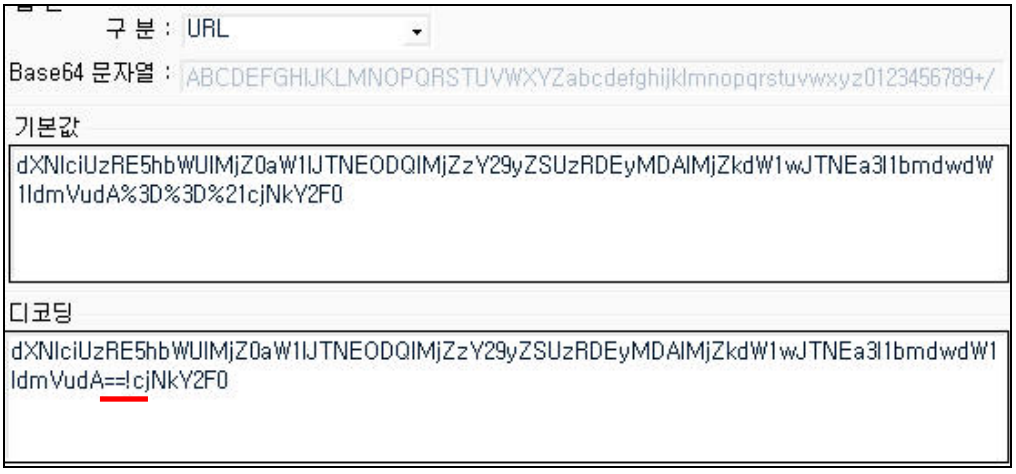


위와 같이 whatisthis 의 값이 평문 형태가 아님을 알 수 있습니다. 그렇지만 유심히 살펴보면 소문자 a-z 그리고 대문자 A-Z와 숫자 0-9로 이루어진 것을 알 수 I 있습니다. BASE64로 인코딩 된 데이터 입니다.

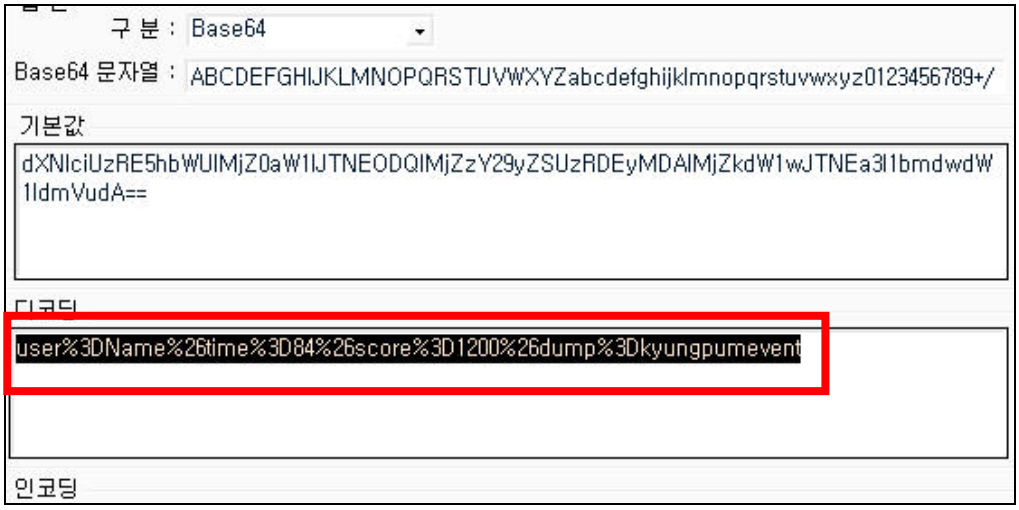
위와 같이 BASE64 부분만 선택하여 decoder에 삽입합니다.
여러가지 디코더가 있지만 전 '방립동'형이 제작한 툴을 사용하도록 하겠습니다.
그렇지만 바로 디코딩을 눌러도 디코딩이 안되고 있습니다.



자세히 살펴보면 %3D%3D%21 으로 URL 인코딩 된 부분이 있습니다.



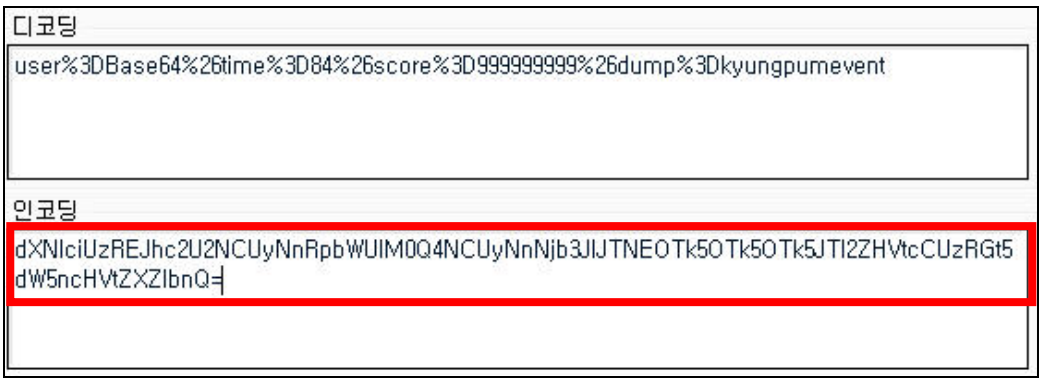
URL 디코딩 해보면 ==! 인것을 알 수 있습니다. !는 BASE64에서 사용되는 문자가 아니므로 구분자로 사용된 부분 입니다. 그 앞까지만 다시 드래그하여 디코딩 하도록 하겠습니다.



위와 같이 디코딩 된 것을 알 수 있습니다. 이번에는 그 값을 조작해서 인코딩 하겠습니다.

user%3DBase64%26time%3D84%26score%3D999999999%26dump%3Dkyungpumevent

위와 같이 데이터를 변조하여 인코딩 하도록 하겠습니다.



이렇게 인코딩 한 값을 다시 Paros에 붙여 넣습니다.

변조한 데이터로 붙여 넣기 한 후에 Continue 버튼을 클릭 합니다.
주의 할 것은 %21 뒤에 있는 구분자를 그대로 입력을 해주셔야 합니다. 그렇지 않으면 전송 데이터 형에 맞지 않기 때문에 반영이 되지 않습니다.

순위	아이디	점수	시간
1	Base64	999999999	84
2	test	900000000	84
3	vit	600000000	59
4	r3dcat	9999999	3

위와 같이 결과가 쉽게 조작되는 것을 알 수 있습니다. 이렇게 해서 IE를 통해서 전송되는 이벤트 게임의 결과를 쉽게 조작해 보았습니다.

3 대처방안 (OWASP Top10 2007 수록내용)

인증은 안전한 통신과 자격 증명 저장에 달려 있다. 우선적으로 SSL은 어플리케이션의 모든 인증된 부분을 위한 유일한 옵션(A9 – 불안전한 통신을 보라)이며 모든 자격 인증이 해시나 암호화 형태(A8 – 불안전한 암호화 저장을 보라)로 저장되도록 확실히 하라.
인증 취약점을 보호하는 것은 주의 깊은 계획을 필요로 한다. 가장 중요한 고려사항은 다음과 같다:

□ 본래의 세션 관리 메커니즘만 사용한다. 어떤 상황 하에서도 이차적인 세션 처리기를 사용하거나 개발해서는 안 된다.

□ URL 또는 요청에서 새롭거나 사전 설정되거나 또는 검증되지 않은 세션 식별자를 허용하면 안 된다. 이것은 세션 고정 공격으로 불린다.

□ "내 계정 정보 기억" 기능 또는 자체 제작된 싱글 사인 온 기능과 같은 인증 및 세션 관리를 위한 여러분의 특별히 개발된 쿠키를 제한하거나 제거한다. 이는 완전히 입증된 SSO나 연합 인증 솔루션에 적용되지 않는다.

□ 적절한 강도와 요소의 수와 함께 싱글 인증 메커니즘을 사용하라. 이 메커니즘이 속임수이거나 재입력 공격을 받지 않도록 조심하라. 이 메커니즘을 지나치게 복잡하게 만들어서 자기 스스로 공격받게 되도록 만들지 말라.

□ 암호화되지 않은 페이지로부터 로그인 과정이 시작되도록 허용하지 말라. 자격 증명이나 세션 도난, 피싱 공격 그리고 세션 고정 공격을 방어하기 위해 암호화된 두 번째 페이지에서 항상 로그인을 시작하라.

□ 성공적인 인증 또는 권한 수준 변경에 대해 새로운 세션을 재생성시키는 것을 고려하라.

□ 모든 페이지에 로그아웃 연결주소를 두어라. 로그아웃 함으로써 모든 서버측 세션 상태와 클라이언트 측 쿠키가 제거되어야 한다. 인간적인 요소를 고려하라: 사용자는 성공적으로 로그아웃 하기 보다는, 단지 탭 또는 윈도우를 닫음으로써 끝낼 것이다.

□ 보호할 데이터의 가치 대비 비활성화된 세션이 자동 로그아웃 되도록 타임아웃 기간을 설정하라.(짧을수록 좋다.)

□ 강력한 부수적인 인증 기능(질문과 답변, 암호 재설정)을 사용하라. 이러한 기능은 사용자명과 암호 또는 토큰이 자격 증명이 되듯 자격 증명이 된다. 내용이 노출된 공격을 막기 위해 답변은 단방향 해쉬를 적용하라.

□ URL이나 로그에서 세션 식별자나 유효한 자격 증명의 어떤 부분이 드러나지 않도록 하라(세션을 재작성하거나 로그 파일에서 사용자의 암호를 저장하지 마라). 사용자가 새 암호로 바꿀 때는 이전 암호를 확인하라.

□ IP주소 또는 주소 범위 마스크, DNS 조회 또는 역 DNS 조회, 참조 헤더 또는 이와 유사한 조작 가능한 자격 증명을 유일한 인증 형태로 의존하지 마라.

□ 비밀번호 재설정하는 방법으로 등록된 이메일 주소(참고자료의RSNAKE01을 참조하라)로 비밀정보를 보내는 것에 주의하라. 접속을 재설정하기 위해 제한된 시간동안 사용 가능한 임의 번호들만 사용하고 비밀번호가 재설정되자마자 추적 이메일을 발송하라. 자동 등록한 사용자들이 자신의 이메일 주소를 변경하는 것에 주의하고 변경을 규정하기 전에 이전 이메일 주소로 메시지를 보내라.

예제

□ http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6145
□ http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6229
□ http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6528

참조

□ CWE: CWE-287 (인증 문제), CWE-522 (불충분하게 보호된 자격 인증), CWE-311 (인증 프로토콜의 재입력 공격), 기타.

☐ WASC 위협 분류:

http://www.webappsec.org/projects/threat/classes/insufficient_authentication.shtml

http://www.webappsec.org/projects/threat/classes/credential_session_prediction.shtml

http://www.webappsec.org/projects/threat/classes/session_fixation.shtml

☐ OWASP, http://www.owasp.org/index.php/Guide_to_Authentication

☐ OWASP, http://www.owasp.org/index.php/Reviewing_Code_for_Authentication

☐ OWASP, http://www.owasp.org/index.php/Testing_for_authentication

☐ RSnake01 – http://ha.ckers.org/blog/20070122/ip-trust-relationships-xss-and-you

문의 및 Q&A

r3dcat@gmail.com