

4월 DreamHack CTF Season 5 Round #8 Write-UP

32기 조현서

지금 다룰 CTF 문제는 DreamHack CTF Season 5 Round #8인 easy-login 문제이다. 일단 4문제당 다 너무 어려워 보이는데, 그나마 접근하기에 웹 해킹 문제가 나올 거 같다고 생각했고, 이름부터가 easy-login이어서 다른 문제들보다 그나마 easy하게 풀 수 있지 않을까 해서 선택했다. 웹 해킹 관련해서 배운 건 연합 스터디때 잠깐 배운 게 전부라 걱정되지만, 할 수 있는 선까지 최대한 노력해서 문제를 꼭 풀고 싶다.

문제 설명



Description



관리자로 로그인하여 플래그를 획득하세요!





플래그 형식은 `DH{...}` 입니다.

 Translate

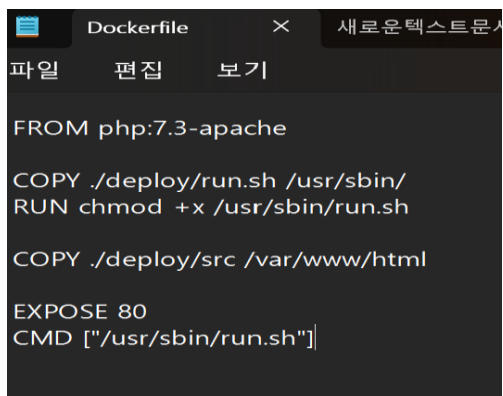
문제 설명을 보면, 관리자로 로그인하여 `DH{...}` 형식으로 되어 있는 플래그를 찾아내는 문제인 것 같다. 문제 설명을 봐도 감이 잘 안 오지만, 일단 문제 파일을 다운 받아서 확인해보고자 한다.

 Dockerfile	2023-12-14 오후 2:56	파일
 deploy	2023-12-14 오후 2:56	파일 폴더

 run.sh	2023-12-14 오후 2:56	SH 파일
 src	2023-12-14 오후 2:56	파일 폴더

 flag.php	2023-12-14 오후 2:56	PHP 파일	1KB
 index.php	2023-12-14 오후 2:56	PHP 파일	3KB
 login.php	2023-12-14 오후 2:56	PHP 파일	1KB
 style.css	2023-12-14 오후 2:56	Cascading Style She...	1KB

파일을 다운받고, 압축을 풀어보니 생각보다 파일들이 많아서 당황스러웠다. 일단 다른 파일들은 이름만 보았을 때는 어떤 역할을 하고, 어떤 내용을 담고 있는지 예측이 살짝 가는데, Dockerfile은 진짜 초면이라서, 인터넷에서 관련 자료를 찾아보았다. 개발한 어플리케이션을 *컨테이너화할 때 생성하는 방법은 3가지 정도인데, 아무것도 존재하지 않는 이미지로 컨테이너를 생성하고, 어플리케이션을 위한 환경을 설정하고, 소스코드 등을 잘 동작하는 것을 확인하고, 컨테이너를 이미지로 커밋하는 것이다. "컨테이너화"란 소프트웨어 코드를 라이브러리와 같은 필수 요소와 함께 패키지에 포함하여 각자의 "컨테이너"로 분리하는 것을 말한다고 한다. 이렇게 컨테이너화된 것은 어떤 환경에서든 해당 환경의 운영 체제와는 상관없이 이동할 수 있고, 실행할 수 있다고 한다. 다시 원래로 돌아와서, 위와 같은 방법을 사용하면, 어플리케이션이 동작하는 환경을 만들기 위해 일일이 수작업으로 패키지를 설치하고, 이를 복사해야 한다고 한다. 하지만, Docker은 이런 복잡하고 귀찮은 과정을 손쉽게 기록하고 수행한다고 한다. 완성된 이미지를 생성하기 위해 컨테이너에 설치해야 하는 패키지, 소스 코드, 명령어, 셸 스크립트 등을 하나의 파일에 기록해두면, 이 파일을 통해 컨테이너에서 작업을 수행한다고 한다. 이러한 작업을 기록한 파일의 이름을 Dockerfile이라고 한다고 한다. 조금 생소한 개념들도 섞여있어서 조사한 내용들이 100%이해되는 것은 아니지만, 완성된 이미지를 생성하려면 명령어가 Dockerfile을 읽어서 생성하는 것? 이라고 이해했다. 뭔가 다른 파일보다 이 파일이 작동되는데 핵심이 되는 애인가? 라는 생각이 들었다. Flag.php는 최종적으로 플래그와 관련된 것을 내포하고 있을 것 같다는 생각이 들고, Login.php는 로그인 화면 페이지를 다루고 있을 것 같다는 생각이 들었다. Style.css를 보고, css가 뭔지 조사해보았다. 조사해보니까, HTML등으로 작성된 문서가 실제로 웹사이트에 표현되는 방법을 정해주는 스타일 시트 언어라고 한다. 이는 웹페이지를 꾸미려고 작성하는 코드라고 한다. 아마 괜한 추측을 하자면, style.css는 이 로그인하는 웹페이지의 디자인과 스타일을 다룬 파일이라고 생각했다.



```
Dockerfile
FROM php:7.3-apache
COPY ./deploy/run.sh /usr/sbin/
RUN chmod +x /usr/sbin/run.sh
COPY ./deploy/src /var/www/html
EXPOSE 80
CMD ["/usr/sbin/run.sh"]
```

우선 Dockerfile을 열어보니, COPY, RUN이라는 명령어와 run.sh와 나머지 파일이 담겨있

는 src가 언급되어 있는 것을 보니, 아까 조사했던 것처럼, 완성된 어플리케이션을 생성하기 위해 다른 쉘 스크립트와 소스코드가 담겨있는 파일들을 복사해서 가져와서, 기록하고 실행하는 파일이지 않을까 추측했다.

```
run.sh  Dockerfile  새로운텍스트문서
파일  편집  보기

#!/bin/bash

export FLAG="test"
&>/dev/null /usr/sbin/apachectl -DFOREGROUND -k start
```

run.sh를 열어보니, export FLAG = "test"라는 부분을 보아, 아마 플래그를 밖으로 내보내는데, 그 플래그를 test로 내보내는 건가? 라고 직관적인 영어단어만 보고 추측했다.

```
flag.php  새로운텍스트문서
파일  편집  보기

<?php
$flag = "testflag";
?>
```

flag.php를 열어봤다. 아마 봤을 때, 자바 스크립트? 가 사용된 것 같은데, 전에 어디서 지나가는 식으로 \$가 변수명 앞에 붙는다는 말을 들었던 적이 있던 것이 기억났다. flag라는 변수에 "testflag"라는 문자열을 저장한 건가? 라고 추측했다.

```
index.php  flag.php  run.sh  Dockerfile  새로운텍스트문서
파일  편집  보기

<?php

function generatePassword($length) {
    $characters = '0123456789abcdef';
    $charactersLength = strlen($characters);
    $pw = "";
    for ($i = 0; $i < $length; $i++) {
        $pw .= $characters[random_int(0, $charactersLength - 1)];
    }
    return $pw;
}

function generateOTP() {
    return 'P' . str_pad(strval(random_int(0, 9999999)), 6, "0", STR_PAD_LEFT);
}

$admin_pw = generatePassword(32);
$otp = generateOTP();

function login() {
    if (!isset($_POST['cred'])) {
        echo "Please login...";
        return;
    }

    if (!($cred = base64_decode($_POST['cred']))) {
        echo "Cred error";
        return;
    }

    if (!($cred = json_decode($cred, true))) {
        echo "Cred error";
        return;
    }

    if (!($cred['id'] && $cred['pw'] && $cred['otp'])) {
        echo "Cred error";
    }
}
```

```
return;
}

if ($cred['id'] != 'admin') {
    echo "Hello," . $cred['id'];
    return;
}

if ($cred['otp'] != $GLOBALS['otp']) {
    echo "OTP fail";
    return;
}

if (!strcmp($cred['pw'], $GLOBALS['admin_pw'])) {
    require_once('flag.php');
    echo "Hello, admin! get the flag: " . $flag;
    return;
}

echo "Password fail";
return;
}

?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <title> Easy Login</title>
</head>
<body>
    <div class="login-container">
        <h2> Login as admin to get flag</h2>
        <form action="login.php" method="post">
            <div class="form-group">
                <label for="id"> ID</label>
                <input type="text" name="id"> </br>
```

```

</div>
<div class="form-group">
  <label for="otp">OTP</label>
  <input type="text" name="otp"> </br>
</div>
<button type="submit" class="button">Login</button>
</form>
<div class="message">
  <?php login(); ?>
</div>
</div>
</body>
</html>

```

index.php를 봤을 때, 처음 <?php ... ?> 부분을 보면, 내가 아는 프로그래밍 언어는 아니지만, 패스워드를 생성하고, 그 패스워드는 0123456789abcdef 를 사용해서만 조합될 수 있게끔 된 것 같다고 생각했다. 그리고 변수 i가 패스워드 길

이보다 작으면, 변수 i가 0부터 시작해서 계속 1씩 커지면서 반복되는데, 0123456789abcdef중에서 랜덤으로 숫자를 뽑아 패스워드로 설정하는 것 같다고 생각했다. 또한, OTP도 생성해내서, 0부터 999999까지의 숫자 중에서 6개를 추출해서 OTP로 설정하는 것 같다고 생각했다. 그리고 뒤에 여러 if문들을 대략적으로 보았을 때, cred이라는 말이 나왔는데, 이게 뭘까 곰곰히 생각했다. 그러다가 최근에 보안뉴스를 작성하면서, 다뤘던 크리덴셜(credential)이 생각났다. 내가 로그인하는 아이디와 패스워드와 관련된 쪽으로 억지로 끼워 맞추는 거일수도 있지만, 그냥 크리덴셜이라고 생각하기로 했다. 그렇게 생각했을 때, 로그인 페이지에서 사용자를 식별하기 위해 요구하는 크리덴셜인 아이디와 패스워드와 OTP의 조건들을 따져, 크리덴셜이 잘못되었다, 다시 로그인해주세요, OTP가 틀렸습니다, 안녕하세요, OO님 등을 출력하는 거 같았다. 뒤에 부분은 지난번에 연합 스터디에서 배웠던 html인 것 같았다. 그래서 그때 배운 것처럼 한 번 확장자명으로 html로 바꿔볼까 했다. (뭐라도 해보자는 마음으로....)



Login as admin to get flag

ID

PW

OTP

바탕화면에 새로 생긴 index.html을 들어가보니, 이런 로그인 화면이 생겼다.... 역시 연합 스터디 때 배운 게 허투루 되지 않았구나 생각이 들었다. 예상 외로 login.php가 아니라

index.php에서 이런 로그인 페이지가 실행되서 신기했다.

```
index.html login. x 음.html flag.php +
파일 편집 보기

<form id="redir" action="index.php" method="post">
  <?php
    $a = array();
    foreach ($_POST as $k => $v) {
      $a[$k] = $v;
    }

    $j = json_encode($a);
    echo <input type="hidden" name="cred" value="". base64_encode($j) . ">";
  ?>
</form>

<script type="text/javascript">
  document.getElementById("redir").submit();
</script>
```

다음은 login.php를 살펴보자. 보면, action="index.php"를 보면 index.php를 동작하라고 하는 건가? 라고 생각했다. 그리고 method="post"를 보고, method가 뭘까 생각해봤는데, 컴정개론 시간에 배운 객체 지향 언어가 생각이 났다. 그 중에 자바가 객체 지향 언어이라고 배웠던 것이 기억이 나면서, 그러면 자바 스크립트도 객체 지향 언어이지 않을까? 라는 생각이 들었다. 그리고 객체는 상태와 행위로 구성되는데, 그렇게 행동하게끔 하는 함수를 메소드라고 했던 것이 생각났다. 그래서 아마 저기서 method는 함수를 나타낸 거지 않을까? 라는 생각과 함께, 그럼 post라는 함수가 있다는 거구나 라고 생각했다.

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.login-container {
  background-color: white;
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.1);
  text-align: center;
}

.login-container h2 {
  margin: 0;
  color: #333;
}

.form-group {
  margin-top: 20px;
  text-align: left;
}

.form-group label {
  display: block;
  margin-bottom: 5px;
}

.form-group input {
  width: 90%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
}

.button {
  background-color: #5cb85c;
  color: white;
  padding: 10px 20px;
  margin-top: 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}

.button:hover {
  background-color: #4cae4c;
}

.message {
  margin-top: 20px;
  color: #777;
}
```

Style.css 파일을 열어보니, 진짜 예상한대로 색깔이나 폰트 크기 등 디자인에 관한 거 같았다.

일단 플래그를 찾으려면, index.php를 중점으로 더 봐야겠다고 생각했다. 그래서 index.html의 소스코드를 중점적으로 보기로 했다.

```
if (!strcmp($cred['pw'], $GLOBALS['admin_pw'])) {  
    require_once('flag.php');  
    echo "Hello, admin! get the flag: " . $flag;  
    return;  
}
```

이 부분이 flag를 출력하는 조건문인 것 같은데, 이 부분을 보면..

Strcmp는 예전에 C++ 과제로, cstring 함수 정리할 때 조사했었는데, 문자열 두 개를 비교하는 함수였다. 물론 C++언어는 아니지만, 비슷한 역할을 하지 않을까 생각하면, 변수 cred에 저장된 pw와 GLOBALS에 저장된 admin_pw를 비교했을 때, !가 있으니까 다를 때 이 조건문이 실행되는 거 아닐까 추측했다. 이 조건이 맞으면, flag.php 파일을 실행시켜, 플래그 값을 출력하는 거일 것 같다고 생각했다. 이 index.php를 다시 보면, function generatePassword(\$length)와 \$admin_pw=generatePassword(32)인 부분을 통해, admin_pw는 패스워드 길이가 32인 것을 생성하는 거이지 않을까 생각했다. 그러면, 랜덤으로 32자리가 생성된 admin_pw와 랜덤으로 생성된 pw의 문자열이 다를 때, flag가 출력되는 건가? 싶은 생각이 들었다.

꼭 풀고 싶었는데, 생각보다 너무 어려운 거 같다. 이렇게 내가 했던 것처럼 얼렁뚱땅 짐작하면서, 접근하는 것이 올바른 접근법인가 의문점이 드는 것 같다. 드림핵에서 보니까, 이 문제가 쉬운 1단계로 지정됐던데 이런 1단계 문제를 풀지 못하는 건가... 싶어서 조금 좌절감이 든다. 하지만, 이번 기회에 앞으로 열심히 공부하고 파고들고 말 것이라는 자극제가 된 것 같다. 조금 더 기본적인 지식을 쌓고, 다른 사람들의 write up을 보면서, 이 문제를 꼭 풀고 싶다는 생각이 들었다.