xxsj_web19

> **Note**
>
> `GZCTF-challenges/xxsj_web19`
>
> 使用了`js/jquery.min.js` 和 `js/crypto-js.js`

> **Tip**
>
> 测试人员可直接进入容器，查看 `/tmp/ctf_aes_cache.txt` 文件查阅信息

1. 按下 `CTRL` + `U` 或用其他方式查看源码

```
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <script src="js/jquery.min.js"></script>
    <script src="js/crypto-js.js"></script>
    <title>登录</title>
</head>
<body>
    <h3>请输入管理员账号密码</h3>
    <form action="#" method="post" id="loginForm">
        用户名: <input type="text" name="username" value="admin" readonly><br>
        密　码: <input type="password" name="pazzword" id="pazzword" placeholder="输入16位密码"><br>
        <button type="button" onclick="checkForm()">提交</button>
    </form>
</body>
<script type="text/javascript">
    var key = "2271508549587588";
    var iv = "XoZfRjqpTQMNNEzt";

    function checkForm() {
```

```
        var pazzword = $("#pazzword").val();
        var passwordRegex = /^[A-Za-z0-9_\-]{16}$/;
        if (!passwordRegex.test(pazzword)) {
            alert("密码必须是16位字符（大小写字母、数字、_、-）！");
            return;
        }
        pazzword = encryptToHex(pazzword, key, iv);
        $("#pazzword").val(pazzword);
        $("#loginForm").submit();
    }

    function encryptToHex(data, key, iv) {
        var key_latin1 = CryptoJS.enc.Latin1.parse(key);
        var key_sha1 = CryptoJS.SHA1(key_latin1);
        var key_sha1_hex = key_sha1.toString(CryptoJS.enc.Hex);
        var key_256_hex = (key_sha1_hex + key_sha1_hex).substring(0,
64);
        var key_256 = CryptoJS.enc.Hex.parse(key_256_hex);

        var iv_latin1 = CryptoJS.enc.Latin1.parse(iv);
        var encrypted = CryptoJS.AES.encrypt(data, key_256, {
            iv: iv_latin1,
            mode: CryptoJS.mode.CBC,
            padding: CryptoJS.pad.ZeroPadding
        });
        return encrypted.ciphertext.toString(CryptoJS.enc.Hex);
    }
</script>

<!--
error_reporting(0);
$flag = getenv('FLAG') ?: 'flag{_Not_here_}';
$u = $_POST['username'];
$p = $_POST['pazzword'];
$correct_pazzword = "97ebaf85373ac6954acaf98ccec94a2d";
if(isset($u) && isset($p)){
    if($u === 'admin' && $p === $correct_pazzword){
        echo $flag;
    }
}
```

```
-->

</html>
```

2. 得到如下信息

```
加密算法：AES
密钥长度：256 位（AES-256）
加密模式：CBC（Cipher Block Chaining）
填充方式：ZeroPadding
密钥派生方式：使用 key 的 latin1 编码，做 SHA1 摘要，得到 hex 字符串，拼接两次取前 64 位，作为 AES-256 密钥
IV（初始化向量）：Latin1 编码字符串
最终密文编码：十六进制（Hex）
```

3. 根据得到的信息编写脚本解密密文

```python
from Crypto.Cipher import AES
from Crypto.Hash import SHA1
from binascii import unhexlify

def sha1_key_256(key):
    key_bytes = key.encode('latin1')  # CryptoJS.enc.Latin1.parse
    sha1 = SHA1.new()
    sha1.update(key_bytes)
    sha1_hex = sha1.hexdigest()
    key_256_hex = (sha1_hex * 2)[:64]
    return bytes.fromhex(key_256_hex)

def decrypt(cipher_hex, key, iv):
    aes_key = sha1_key_256(key)
    iv_bytes = iv.encode('latin1')
    cipher = AES.new(aes_key, AES.MODE_CBC, iv=iv_bytes)
    cipher_bytes = unhexlify(cipher_hex)
    plain = cipher.decrypt(cipher_bytes)
    # Remove ZeroPadding manually
    plain = plain.rstrip(b'\x00')
    return plain.decode()

if __name__ == "__main__":
```

```
# ------------------------------------------------------------------------
--------
# 请将下列值替换为实际值
    key = "2271508549587588"
    iv = "XoZfRjqpTQMNNEzt"
    cipher_hex = "97ebaf85373ac6954acaf98ccec94a2d"
# ------------------------------------------------------------------------
--------
    password = decrypt(cipher_hex, key, iv)
    print("密码:", password)
```

得到输出结果如下：

```
密码: TRrR7AYADzs45Nvz
```

4. 输入密码，点击`提交`，获得`FLAG`