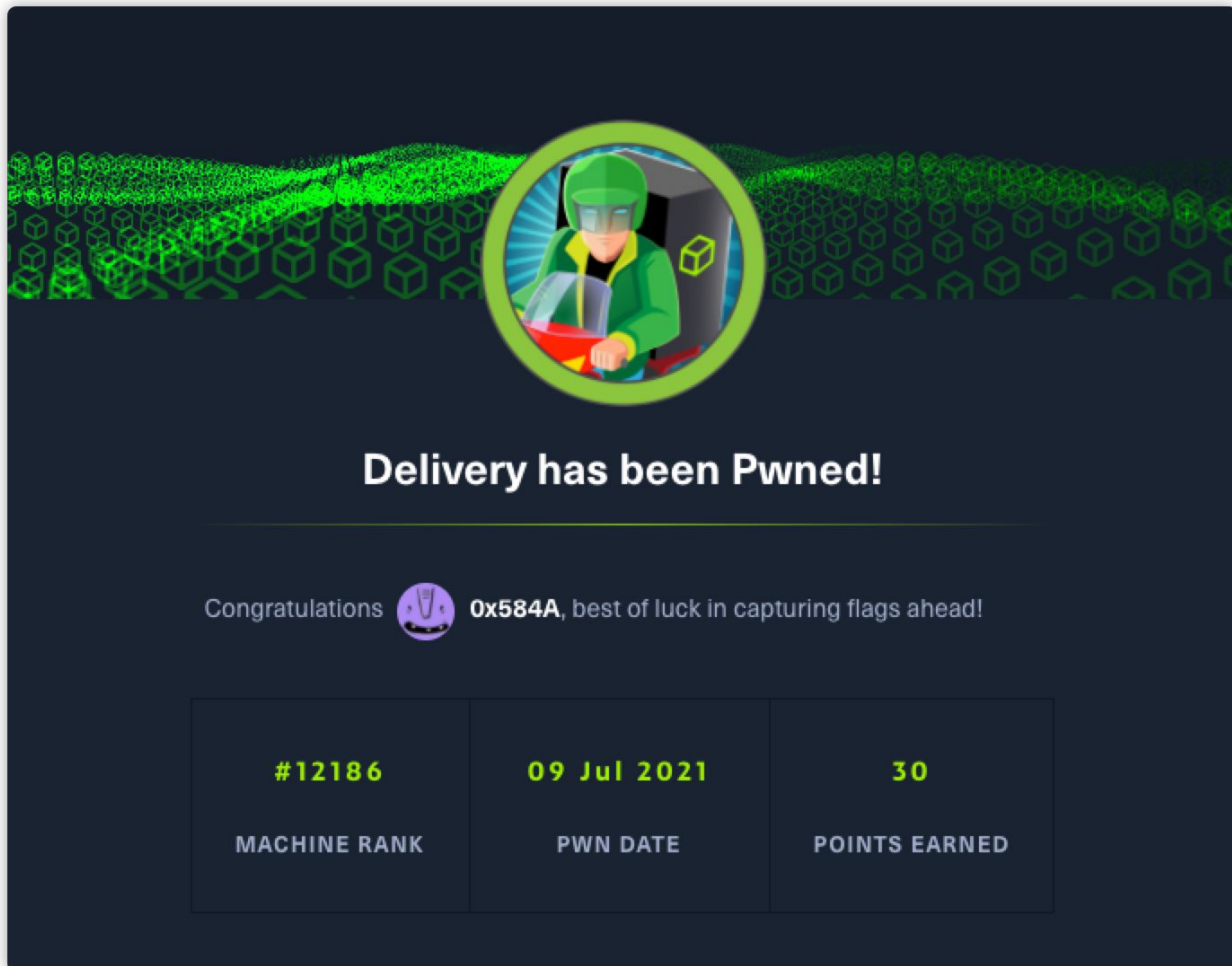


[概述 \(Overview\)](#)  
[攻击链 \(Killchain\)](#)  
[枚举 \(Enumeration\)](#)  
[立足点 \(Foothold\)](#)  
[权限提升 \(Privilege Escalation\)](#)  
[参考](#)

## 概述 (Overview)



时间: 2021-07-09

机器作者: ippsec

困难程度: easy

描述: 考察信息收集能力及如何通过 Ticket Trick 进行攻击的实例，最后通过已知的信息枚举字典。

Flags: User: <md5> , Root: <md5>

### INFORMATION:

- Web
- Bash
- Account Misconfiguration

## 攻击链 (Killchain)

通过使用 nmap 识别获得开发的HTTP服务信息，通过查看页面的版本的信息识别部署的服务。通过在 osTicket 上创建一个新的功能，获得系统自动分配的站点邮箱，随后利用该邮箱注册 Mattermost 系统的登录账号，最后

通过 osTicket 的工单信息查询接收 Mattermost 系统发来的邮箱激活信息，成功加入到目标系统的聊天组中。通过联通组中的历史信息，用该账号密码成功 mattermost 用户ssh。

最后通过对历史信息的查看，使用 hashcat 的 best64.rule 规则生成新增的密码字典，成功枚举出MYSQL中保存的 root 用户明文密码，成功得到 root flag。

## 枚举（Enumeration）

老规矩，Nmap 开局对目标开发端口及端口进行扫描识别：

```
1 Running a full scan on 10.10.10.222
2
3 PORT      STATE SERVICE
4 22/tcp    open  ssh
5 80/tcp    open  http
6 8065/tcp  open  unknown
7
8 PORT      STATE SERVICE VERSION
9 8065/tcp  open  unknown
10 | fingerprint-strings:
11 |   GenericLines, Help, RTSPRequest, SSLSessionReq, TerminalServerCookie:
12 |     HTTP/1.1 400 Bad Request
13 |     Content-Type: text/plain; charset=utf-8
14 |     Connection: close
15 |     Request
16 |   GetRequest:
17 |     HTTP/1.0 200 OK
18 |     Accept-Ranges: bytes
19 |     Cache-Control: no-cache, max-age=31556926, public
20 |     Content-Length: 3108
21 |     Content-Security-Policy: frame-ancestors 'self'; script-src 'self' cdn.rudderlabs.c
22 |     Content-Type: text/html; charset=utf-8
23 |     Last-Modified: Fri, 09 Jul 2021 06:27:05 GMT
24 |     X-Frame-Options: SAMEORIGIN
25 |     X-Request-Id: uabw1n4qej8xtficdbjxqx5isw
26 |     X-Version-Id: 5.30.0.5.30.1.57fb31b889bf81d99d8af8176d4bbaaa.false
27 |     Date: Fri, 09 Jul 2021 06:31:46 GMT
28 |     <!doctype html><html lang="en"><head><meta charset="utf-8"><meta name="viewport" cc
29 |   HTTPOptions:
30 |     HTTP/1.0 405 Method Not Allowed
31 |     Date: Fri, 09 Jul 2021 06:31:47 GMT
32 |_    Content-Length: 0
```

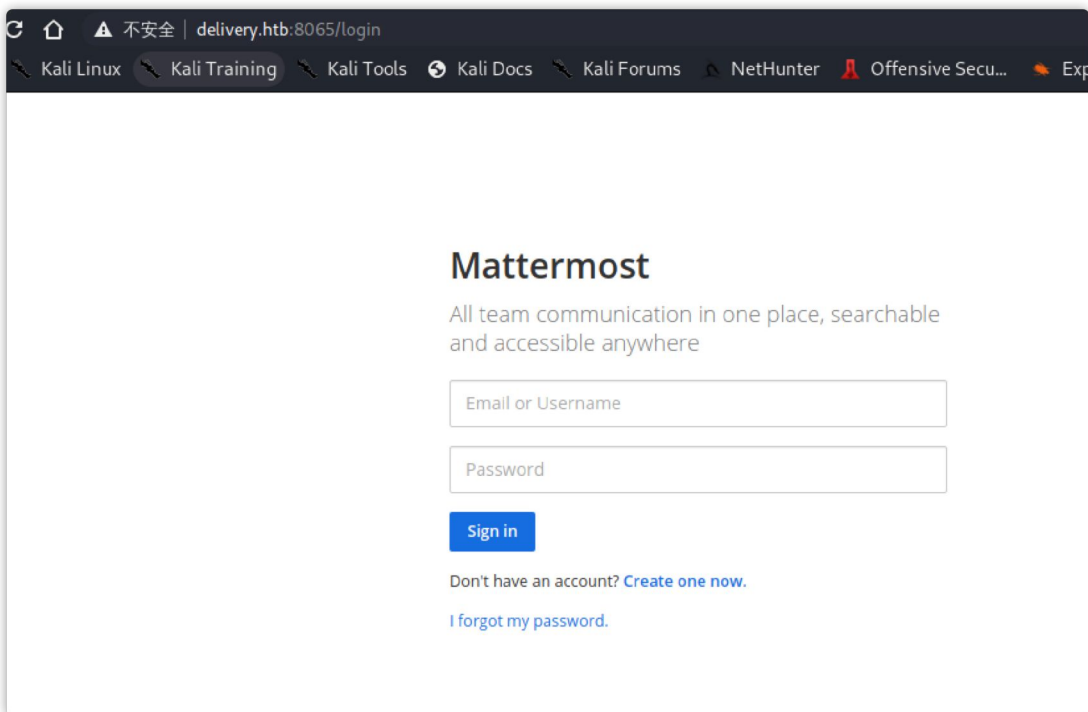
使用浏览器查看目标服务器的HTTP服务，在页面源码发现中发现新的域名：

```
24         <span class="icon fa-diamond"></span>
25     </div>
26     <div class="content">
27         <div class="inner">
28             <h1>Delivery</h1>
29             <p>!--[-->The best place to get all your email related support <!--]<br />
30             <!--[-->For an account check out our <a href="http://helpdesk.delivery.htb">helpdesk</a>!--]</p>
31         </div>
32     </div>
33     <nav>
34         <ul>
35             <li><a href="#contact-us">Contact Us</a></li>
36             <!--<li><a href="#elements">Elements</a></li>-->
37         </ul>
38     </nav>
```

将其加入到 hosts 文件中进行访问：

```
1 10.10.10.222 helpdesk.delivery.htb delivery.htb
```

带域名访问 8065 端口看到了一个新的页面，需要输入账号密码用于登录，但很显然我们现在并不知道。



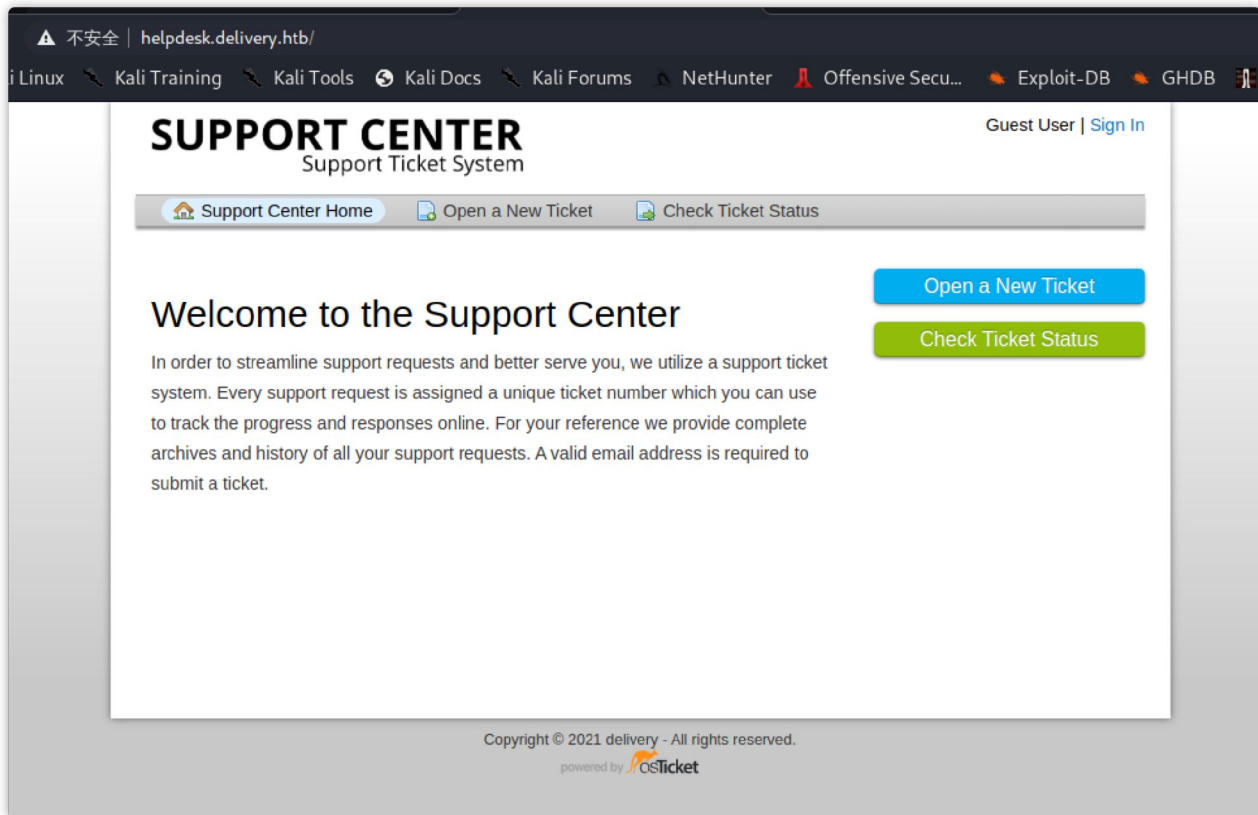
通过搜索 **mattermost** 信息，找到了一个类似的服务部署文档。看样子 **mattermost** 是一个开源的消息通知\聊天服务：

<https://support.websoft9.com/docs/mattermost/stack-installation.html#mattermost-installation-wizard>

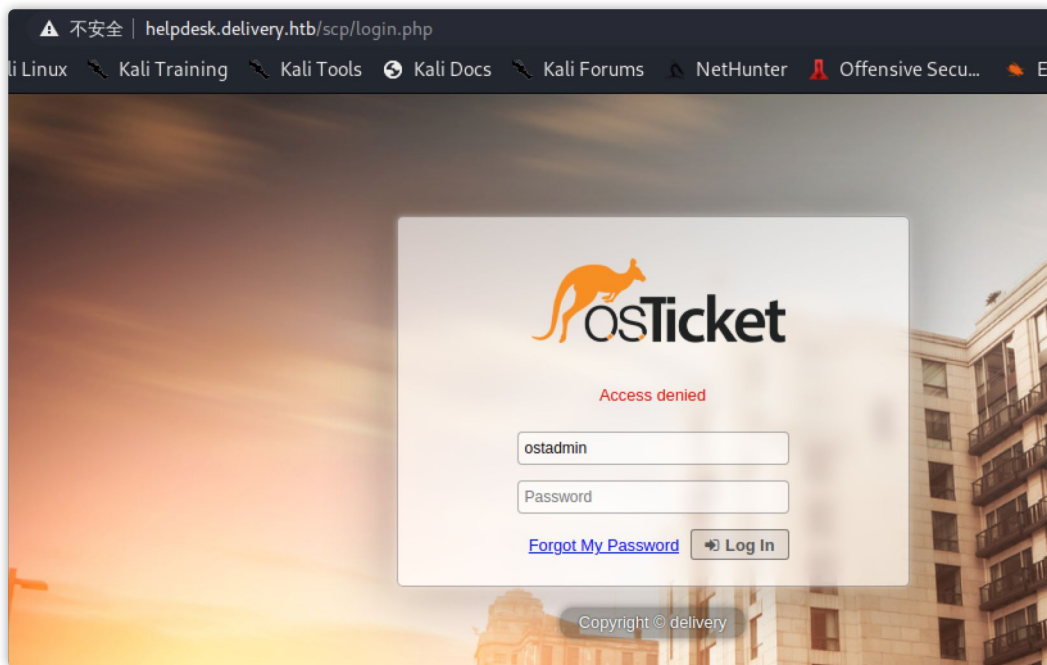
<https://docs.mattermost.com/>

页面上有个 **Create one now** 新建账号，但提交后会提示需要接收邮箱信息进行确认才能激活，尝试后暂时放下。

在查看下另外一个域名，通过页尾信息可以获悉服务是 **osticket**（开源的工单系统）：



尝试搜索 `osticket default login`，使用默认的账号密码发现并不能登录。通过 `dirsearch` 搜索发现 `scp/login.php` 访问后发现是另一个登录入口暂时清楚是用来做什么的：



## 立足点（Foothold）

在 `helpdesk.delivery.htb` 域名中上选择 `Open a New Ticket` 将会看到一个注册页面，通过手册（<https://docs.osticket.com/en/latest/User/Ticket/Open%20A%20Ticket.html>）了解到，这是用来创建新的工单的。

不安全 | helpdesk.delivery.htb/open.php

Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Secu... Exploit-DB GHDB

Please fill in the form below to open a new ticket.

---

**Contact Information**

**Email Address \***

**Full Name \***

Phone Number  
 Ext:

---

**Help Topic**

---

**Ticket Details**  
Please Describe Your Issue

**Issue Summary \***


<> ¶ Aa B / U ↵ ☰ 📷 📺 ☰ 🔗 —

test1

unsaved

Drop files here or choose them

Issue Details is a required field

CAPTCHA Text:   Enter the text shown on the image. \* Please re-enter the text again

当创建完成后，将会被重定向到创建工单的请求确认页面：

Mattermost delivery

全 | helpdesk.delivery.htb/open.php

Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Secu... B

# SUPPORT CENTER

Support Ticket System

Support Center Home Open a New Ticket Check Ticket Status

Support ticket request created

test1,

You may check the status of your ticket, by navigating to the Check Status page using ticket id: 9768739.

If you want to add more information to your ticket, just email 9768739@delivery.htb.

Thanks,

Support Team

这里的提示的信息和文档中的不太一样，存在一个 `9768739@delivery.htb` 邮箱。意思是告诉我可以通过工单ID：9768739 跳转到状态检查页面，如果有跟多信息需要补充，只需要发邮件至 `9768739@delivery.htb`。

点击 `Check Ticket Status` 后输入 email 和 工单ID 可以看到之前提交的工单信息。

还记得 `delivery.htb` 里的 `Create one now` 吗？尝试使用上面得到的邮箱地址进行账号注册：

```
1 email address:9768739@delivery.htb
2 username:test1
```

3 password:..QWER!@#\$56

再次回到 **Check Ticket Status** 页面，提交表单后就看到了带有注册成功的验证连接消息：

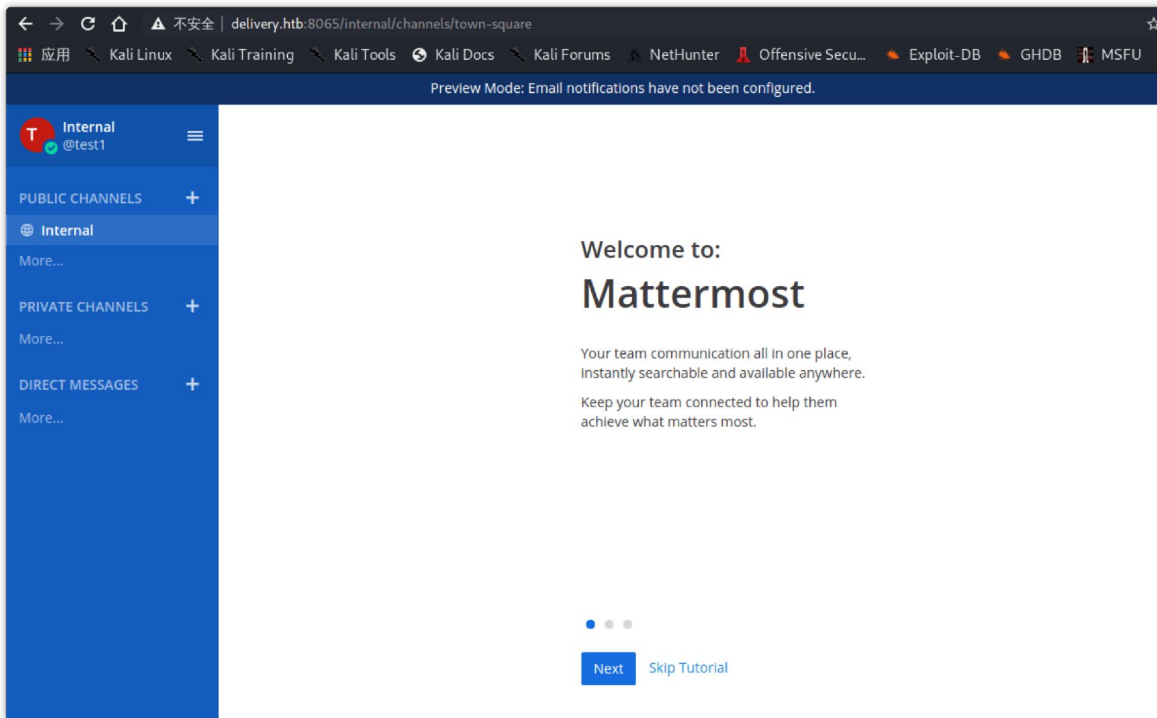
The screenshot shows a web browser window with the URL `helpdesk.delivery.htb/tickets.php?id=7`. The page is titled "SUPPORT CENTER" and "Support Ticket System". It features a navigation bar with links like "Support Center Home", "Open a New Ticket", and "View Ticket Thread". A message prompts the user to "Looking for your other tickets? Sign In or register for an account". The ticket details for "test1 #9768739" are displayed, including "Basic Ticket Information" (Status: Open, Department: Support, Create Date: 7/9/21 3:39 AM) and "User Information" (Name: Test1, Email: test1@test.com, Phone: ). A post from "test1" dated 7/9/21 3:39 AM contains the message: "---- Registration Successful ---- Please activate your email by going to: [http://delivery.htb:8065/do\\_verify\\_email?token=uqgxh7jw49wdp76s8hsoojdq5nqje3ey4p6qi9ed4emmhnoarehz9traqhwca1ae&email=9768739%40delivery.htb](http://delivery.htb:8065/do_verify_email?token=uqgxh7jw49wdp76s8hsoojdq5nqje3ey4p6qi9ed4emmhnoarehz9traqhwca1ae&email=9768739%40delivery.htb)". Below the post is a "Post a Reply" section with a rich text editor.

浏览地址将提示账号注册成功的提示：

The screenshot shows a web browser window with the URL `delivery.htb:8065/should_verify_email?email=9768739%40delivery.htb&redirect_to=/login.php`. The page has a dark header with navigation links. The main content area is white and features the heading "Mattermost: You are almost done". Below the heading, it says "Please verify your email address. Check your inbox for an email." and includes a "Resend Email" button. At the bottom, a green box with a checkmark icon and the text "Verification email sent." indicates the email has been successfully sent.

当然我们输入注册的账号密码后，成功登录该系统：

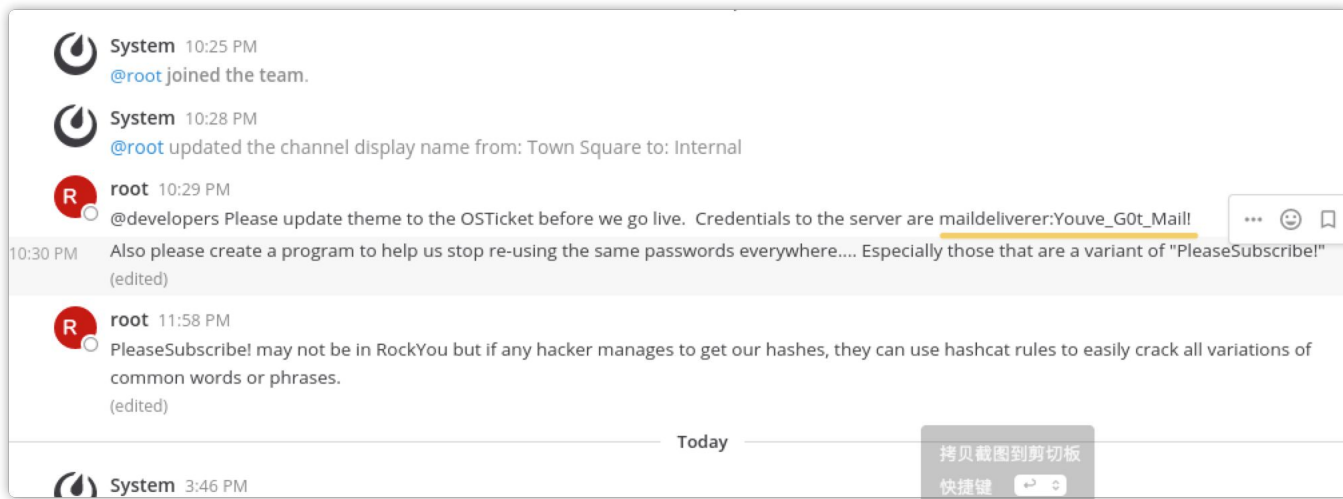




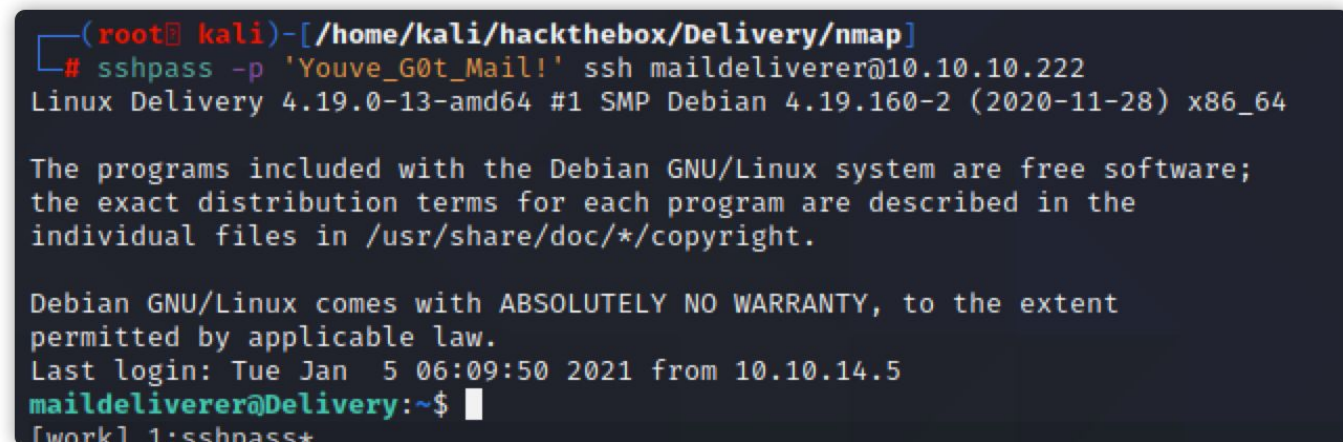
看懂了吗？解释下：

- 首先我们是没有一个能接收目标站点邮箱验证的邮箱，但我们在子域名工单系统中发起工单时，它会为我们分配一个邮件。
- 同时，通过分配的邮箱加分配的工单ID可以查看到邮件内的消息。
- 这样组合起来我们就可以用工单分配的邮箱来接收注册验证信息。

在系统内翻一翻，找到了 **root** 用户在组里发布的消息：



从消息中我们得到了一组密码：**maildeliverer:Youve\_G0t\_Mail!**，使用该密码成功登录目标服务器，得到 user flag。



# 权限提升 (Privilege Escalation)

传递 `linpeas.sh` 进行深度的信息收集，发现还存在一个 `mattermost` 的用户：

```
root:x:0:0:root:/root:/bin/bash

[+] Users with console
maildeliverer:x:1000:1000:MailDeliverer,,,:/home/maildeliverer:/bin/bash
mattermost:x:998:998::/home/mattermost:/bin/sh
root:x:0:0:root:/root:/bin/bash

[+] All users & groups
uid 0(root), gid 0(root), groups 0(root)
```

在进程信息中查看到存在一个同名的进程在执行：

```
mysql      625  0.1  2.6 1719020 107680 ?        Ssl  02:27   0:00 /usr/sbin/mysqld
root       636  0.0  0.1  29208  8016 ?        Ss   02:27   0:00 /usr/sbin/cupsd -l
matterm+   694  0.3  4.2 1797732 169856 ?        Ssl  02:27   0:20 /opt/mattermost/bin/mattermost
matterm+   898  0.0  0.3 1234164 15936 ?        Sl   02:30   0:00 _plugins/com.mattermost.plugin-channel-export/server/dist/plugin-linux-amd64
matterm+   906  0.0  0.5 1239060 23824 ?        Sl   02:30   0:00 _plugins/com.mattermost.nps/server/dist/plugin-linux-amd64
root       871  0.0  0.4  29532 18312 ?        S    02:28   0:00 python3 /root/py-smtp.py
maildel+  1413  0.0  0.2  21144  9044 ?        Ss   03:49   0:00 /lib/systemd/systemd --user
maildel+  1415  0.0  0.0 105092  2344 ?        S    03:49   0:00 _ (sd-pam)
```

接着在 `/opt/mattermost/config/config.json` 文件中获得到 mysql 数据库连接账号密码：

```
1 "SqlSettings": {
2     "DriverName": "mysql",
3     "DataSource": "mmuser:Crack_The_MM_Admin_PW@tcp(127.0.0.1:3306)/mattermost?charset=utf8mb4",
4     ... snip ...
}
```

随后在数据库中查询到含有 root 账号的密码表：

```
1 MariaDB [mattermost]> select username,Password from Users;
2 +-----+-----+
3 | username                               | Password |
4 +-----+-----+
5 | surveybot                             |          |
6 | c3ecacacc7b94f909d04dbfd308a9b93     | $2a$10$u5815SIBe2Fq1FZlv9S8I.VjU3zeSPBrIEg9wvpiLaS7 |
7 | 5b785171bf34762a933e127630c4860     | $2a$10$3m0quqyvCE8Z/R1gFcCOW06tEj6FtqtBn8fRAXQXmaKm |
8 | root                                  | $2a$10$VM6EeymRxJ29r8Wjkr8Dtev00.1StWb4.4ScG.anuu7v |
9 | ff0a21fc6fc2488195e16ea854c963ee     | $2a$10$RnJsISTLc9W3iUcUggl1K0G9vqADED24CQcQ8zvUm1Ir |
10 | channelexport                         |          |
11 | test1                                 | $2a$10$jWk9yKa.wMx4TbmyI2iirOY3t.HmVnJsvA.ZVZHYRU8R |
12 | 9ecfb4be145d47fda0724f697f35ffaf     | $2a$10$s.cLPsjAVgawG0JwB7vrqenPg2lrDt0ECRtjwWah0zHf |
13 | test                                  | $2a$10$ZdsNcAb3Vy7DmP3xfXVCf0jINTt0ndgtYHHwfJ77NX9c |
14 +-----+-----+
15 9 rows in set (0.001 sec)
```

但是我花了大量的时间去解密，字典都跑完了还是没解出来。怀疑是不是思路错了，最后在论坛里找到了有点有用的信息：





CONFIANT

February 23 Report Spoiler

```
root@Delivery:~# id ; hostname
uid=0(root) gid=0(root) groups=0(root)
Delivery
```

finally rooted.

foothold:

- As everyone said: follow the hints.
- Do it in the right order.
- find a way to "verify" locally

User:

after you get the email play around there and you find it.

root:

- first step is enumeration
- second step use a tool mentioned in the hints you were given previously

Seek knowledge from the cradle to the grave

这里说要注意提示中提到的工具，然后我又去看看了之前 root 发过的信息，最后一段中提到了 **hashcat** 还有单词的变体：

PleaseSubscribe! may not be in RockYou but if any hacker manages to get our hashes, they can use hashcat rules to easily crack all variations of common words or phrases.

请订阅！可能不在 RockYou 中，但如果任何黑客设法获得我们的哈希值，他们可以使用 hashcat 规则轻松破解常见单词或短语的所有变体。

也就是说我们需要更具 **PleaseSubscribe!** 去生成常用单词，然后在尝试破解加密才行。

最终在文章 <https://darkless.cn/2019/12/26/hashcat-rule/> 中找到了提示，使用 **best64.rule** 规则生成新字典：

```
echo 'PleaseSubscribe!' | hashcat -r /usr/share/hashcat/rules/best64.rule -o password.txt --stdout
```

```
(root@kali)-[/home/kali/hackthebox/Delivery/file]
# john ./root_hash.txt --wordlist=./password.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
PleaseSubscribe!21 (?)
1g 0:00:00:00 DONE (2021-07-09 18:29) 2.083g/s 75.00p/s 75.00c/s 75.00C/s PleaseSubscribe!..PleaseSubscrio
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

成功解出密码哈希的明文为：**PleaseSubscribe!21**，获得 root flag。

```
MariaDB [mattermost]>
MariaDB [mattermost]> exit
Bye
maildeliverer@Delivery:/opt/mattermost$ su root -
Password:
root@Delivery:/opt/mattermost# █
[work] 1:sshpass*
```

有意思的是，在root用户下ippsec还留有一段话，注明制作该box的灵感来源。

```
root@Delivery:~# cat note.txt
I hope you enjoyed this box, the attack may seem silly but it demonstrates a pretty high risk vulnerability I've seen several times. The inspiration for the box is here:

- https://medium.com/intigriti/how-i-hacked-hundreds-of-companies-through-their-helpdesk-b7680ddc2d4c

Keep on hacking! And please don't forget to subscribe to all the security streamers out there.

- ippsec
root@Delivery:~#
```



## 微信搜一搜

🔍 一个人的安全笔记

### 参考

- <https://medium.com/intigriti/how-i-hacked-hundreds-of-companies-through-their-helpdesk-b7680ddc2d4c>
- <https://darkless.cn/2019/12/26/hashcat-rule/>