How to Samba

Challenge description

You are employed as a system administrator at the “Oxfredy” university. To share lesson files with students, Oxfredy established an SMB service in their network.

Yesterday, a lecturer in Oxfredy tried to upload lesson files through the SMB service, but without success.

Your task is to identify the most likely cause of the upload failure. You must make sure that the SMB service is restricted to authorized users only, to protect Oxfredy.

Your goals

For the investigation, you were provided with the user and password ‘smbuser:0xfre3dy’ to Debian SMB server in the same local network as the lecturer’s user Ubuntu machine.

- Identify potential misconfigurations
- Configure the SMB service properly
- Read the data in the shared file.

My Process:

Start the challenge. I see that there are two different machines, lecturer and admin.

```
lecturer@Ubuntu:~$
```

```
smbadmin@smb-srv:~$
```

I only used the **smbadmin@smb-srv** machine in this challenge.

Within the **smbadmin@smb-srv** machine, I wanted to find out: what privileges do I have? So I entered **‘sudo -l**’ (it’s tricky to distinguish some sans serif fonts, so that’s a lowercase elle I typed in, and that will list the sudo privileges for the **smb-srv** user)

```
smbadmin@smb-srv:~$ sudo -l
Matching Defaults entries for smbadmin on smb-srv:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User smbadmin may run the following commands on smb-srv:
    (root) NOPASSWD: /usr/sbin/service smbd restart
    (root) NOPASSWD: /usr/sbin/service smbd stop
    (root) NOPASSWD: /usr/sbin/service smbd start
    (root) NOPASSWD: /usr/sbin/service smbd status
smbadmin@smb-srv:~$
```

I see we have:

- smbd restart
- smbd stop
- smbd start
- smbd status

So I tried the smbd restart service, but it fails. (*** Starting SMB/CIFS daemon smdb [fail]**)

```
smbadmin@smb-srv:/etc/samba$ sudo /usr/sbin/service smbd restart
 * Stopping SMB/CIFS daemon smbd                                                [ OK ]
 * Starting SMB/CIFS daemon smbd                                                [fail]
smbadmin@smb-srv:/etc/samba$
```

To start my investigation I went to find the Samba configuration (which is located in /etc/samba/)

```
smbadmin@smb-srv:~$ cd /etc/samba/
smbadmin@smb-srv:/etc/samba$
```

And I ran an **ls** (elle ess) to confirm.

```
smbadmin@smb-srv:/etc/samba$ ls
gdbcommands  smb.conf  tls
smbadmin@smb-srv:/etc/samba$
```

I entered **nano smb.conf**

```
smbadmin@smb-srv:/etc/samba$ nano smb.conf
```

My goal here is to look at potential misconfigurations. I started with **Authentication**

```
####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
   server min protocol = SMB3
```

I see the **server min protocol = SMB3**, which only allows SMB version 3 or newer, which is encrypted and much more secure. It's the minimum SMB protocol version that a Samba server will accept. However, maybe I need to enable this protocol to allow Legacy devices or old Windows systems. So, I changed that from SMB3 to **lanman1**, which tells Samba to allow very old SMB protocols, all the way back to LANMAN/SMB1. It'll enable compatibility with very old Windows systems and Legacy devices.

!!! However, be aware that SMB1/LANMAN is unencrypted and vulnerable. Remember, SMB1 was exploited by **WannaCry** and **EternalBlue**. But, since we're in an instructional environment I'm thinking that SMB1 is OK to use, so I proceeded with my changes.

DO NOT use SMB1 in a real production environment.

So, for this challenge in a virtual environment, I set **server min protocol = lanman1**

```
####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
   server min protocol = lanman1
```

Then, I went further along through smb.conf to find what the security settings were. I see its setup with **security = share**. Again, we're in a virtual lab environment so it's OK to change this, but if I were in a production environment I need to be aware that security = share is insecure because it allows anonymous access, so it doesn't identify the individual. I was pretty sure that THIS protocol setting is what had been preventing the service from properly restarting.

```
# Setup usershare options to enable non-root users to share folders
# with the net usershare command.
   security = share
```

So I added the **#** to comment that line out.

```
# Setup usershare options to enable non-root users to share folders
# with the net usershare command.
#  security = share
```

Here I see the comment that "this might need tweaking when using external authentication schemes". I know **%S** is a Samba variable which expands to the share name. So I set this to hard-code access to one specific user, our smbuser.

```
# This might need tweaking when using external authentication schemes
  # valid users = %S
```

Remove the **#**, take out that **%S** and change it to **valid users = smbuser**.

```
# This might need tweaking when using external authentication schemes
    valid users = smbuser
```

Then, I went to the **lessons section** and investigated those settings.
The **path = /smbuser/share** and **read only = yes**. These settings point to a specific directory and the read only set to yes means that users can only view files, not allowing any creating, editing, or deleting. This only shows users whatever is in the /smbuser/share/ location but doesn't let them change anything.

```
[lessons]
 comment = Lesson files share
 path = /smbuser/share/
 acl check permissions = yes
 browsable = yes
 read only = yes
 guest ok = yes
```

So, I made some changes here. I changed the path so that users can see the correct content and let them read, write and modify or delete files (if they're authorized, like smbuser is).
Change: **path = /home/smbuser/share/** and **read only = no**

```
[lessons]
 comment = Lesson files share
 path = /home/smbuser/share/
 acl check permissions = yes
 browsable = yes
 read only = no
 guest ok = yes
```

Now I saved (ctrl X, Y, enter) all my changes.

Now that I edited the smb.conf file, the Samba service will continue to use the previous configuration, so I need to restart Samba. I should be able to do that now because I've modified that earlier step by commenting out **# security = share**.

Run **sudo service smbd restart** to force Samba to reload my updated configuration. I'm expecting an **[OK]** to signify that it runs just fine.

```
smbadmin@smb-srv:~$ sudo service smbd restart
 * Stopping SMB/CIFS daemon smbd                                    [ OK ]
 * Starting SMB/CIFS daemon smbd                                    [ OK ]
```

Woot! It worked.

Now if I list our contents I expect to see **lesson-11.txt**.
When I **cat** that I see our flag for the challenge! (all the way to the right, 33e2bb…)

```
smbadmin@smb-srv:~$ cd /home/smbuser/share/
smbadmin@smb-srv:/home/smbuser/share$ ls
lesson-11.txt
smbadmin@smb-srv:/home/smbuser/share$ cat lesson-11.txt
Subject: Lesson 11 - Definition of planets

This lesson will discuss about the solar system planets and its relation to the code 33e2bb43ea4e40fa8259b203c46e418a
smbadmin@smb-srv:/home/smbuser/share$
```