Samba

Aidan Fetter
10/16/2023

```
This document covers Samba and how to connect it to local
operating systems. You will need either two VMs for this or one
VM and one local machine (assuming they are connected to the same
NAT network).
```

1. What is the SMB protocol?

The Server Message Block (SMB) protocol is a TCP protocol created by Microsoft used to share access to files and printers across different nodes on and across (a) network(s), provide network browsing, and provide an authentication inter-process communication (IPC) mechanism.

- SMB now works on top of the TCP stack using port 445 but was originally used on NetBIOS (Network Basic Input/Output System) using ports 137-139
- Major vulnerability from SMBv1 – WannaCry ransomware (EternalBlue vulnerability)
- In a Microsoft Windows OS, SMB is implanted using the “LanmanServer” service, which is in charge of sharing the resources across the network, and the “LamanWorkstation” service, which is in charge of maintaining the host machine’s name on the network
- Uses the Kerberos protocol (tickets) for larger networks and NTLM (New Technology LAN Manager) for smaller networks
  - Kerberos is protected by eavesdropping and replay attacks and uses symmetric-key cryptography (but may choose to use public-key cryptography)
- Newer versions of Samba use digitally signed messages to prevent against MITM attacks (default = do not allow unsigned connections)
- Up-to-date version: Samba v4.19

2. What is Samba?

- Samba is a software based off of the SMB protocol that allows Unix-like operating systems to communicate with Windows operating systems
- It also provides file and print services for Microsoft Windows clients using a Unix-based operating system
  - Netsmb is the SMB software uqsed for BSD systems (including macOS)

3. Samba Installation and Configuration

3.1. Install Samba and check for successful installation

Commands:

1. sudo apt install samba
2. Check if the installation was successful: whereis samba
3. Check to see if nmbd (the service that runs Samba) is configured: sudo systemctl status nmbd

3.2. Create a folder (or multiple folders) containing one or more files to share

Commands:

1. mkdir nameOfDir
2. cd nameOfDir && touch {a..z}.txt (or whatever command you want to use to add files to the directory)

3.3. Edit Samba configuration files

The configuration file is located in "/etc/samba/". Using whatever text editor you choose (vim, vi, nano, etc.) with root permissions, do the following:

1. Go to the bottom of the file and add the following lines:

```
[nameOfDir]
   comment = Samba in Linux #A description of the directory
   path = /home/user/nameOfDir #Path to your share directory
   read only = yes #The contents of the folder cannot be modified
   browsable = yes #Allows file managers to list the shared
directory under "Network"
```

3.4. Update your firewall to allow Samba

- Using ufw: sudo ufw allow samba
- Using firewall-cmd: sudo firewall-cmd --permanent --add-service=samba
- sudo firewall-cmd –reload

3.5. Add a new user to the Samba server

- sudo smbpasswd -a <name>
  - "<name>" should be the name of your user in your VM. For example, if your home directory contains the directory "user1", the command would be "sudo smbpasswd -a user1
  - You will then be prompted with a password. This can be any password you want and does not have to be the one you use for your VM. However, be sure to remember it as you will need it later when trying to access the shared files.

3.6. Start Samba and ensure connectivity

1. Start smb: sudo systemctl start smdb
   OR
   sudo systemctl start smbd
   - The two commands above start it on your current session
   - To enable it upon system startup: sudo systemctl enable smdb
     - Side note: Using "smb" instead of "smbd" also works. However, this document will use the latter for convenience
2. Start nmbd: sudo systemctl start nmbd

OR
sudo service nmbd start

3.7. Check if the two services are running properly:

- For smbd: sudo systemctl status smbd
  OR
  sudo service smbd status
- For nmbd: sudo systemctl status nmbd
  OR
  sudo service nmbd status

4. View shared files in Windows VM

4.1. Setting up network configurations

1. Locate your Windows VM and edit the "Network" settings to ensure that each OS resides on the same network. I did this by creating a new NAT network (essentially a LAN for VMs to interact with each other on their own network) and attaching it to each of the VMs). You can do this by clicking "Files" → "Tools" → "Network Manager" → "NAT Networks" → "Add".

   - If you're using VirtualBox or VMWare:

     When attaching the NAT network to each VM (or one if you chose to use your local host OS), ensure that "Promiscuous Mode" is set to either "Allow VMs" or "Allow All". This allows the VMs to "speak" to each other while they're on. You may need to save the state of the VM running the Samba server if needed; however, you should be able to change it while it's running.

2. Ensure that both smbd and nmbd are active and running on the Samba-hosting system. If you previously closed the VM, you might need to restart both services by typing "sudo systemctl start smbd" and sudo systemctl start nmbd".

3. Open the Windows VM and open the command line on both VMs. Run the command "ifconfig" on the Linux VM and "ipconfig" on the Windows VM to get their corresponding IP addresses. After obtaining this, ping the Linux machine from the Windows machine to ensure they are connected successfully (and vice versa). If not, go back to step 1 and review.

4. If both machines are connected to each other, press the Windows Key + R in the Windows VM to open up the Run application, and type "\\IP\Share", where "IP" is the IP address of the Linux machine and "Share" is the shared folder you created earlier. After pressing enter, you should now have access to the shared files after entering your credentials!

   - Side note: You can also run "\\IP" and the share will show up as the directory rather than the contents of the directory.

4.2. If the IP address cannot be found and both machines are connected

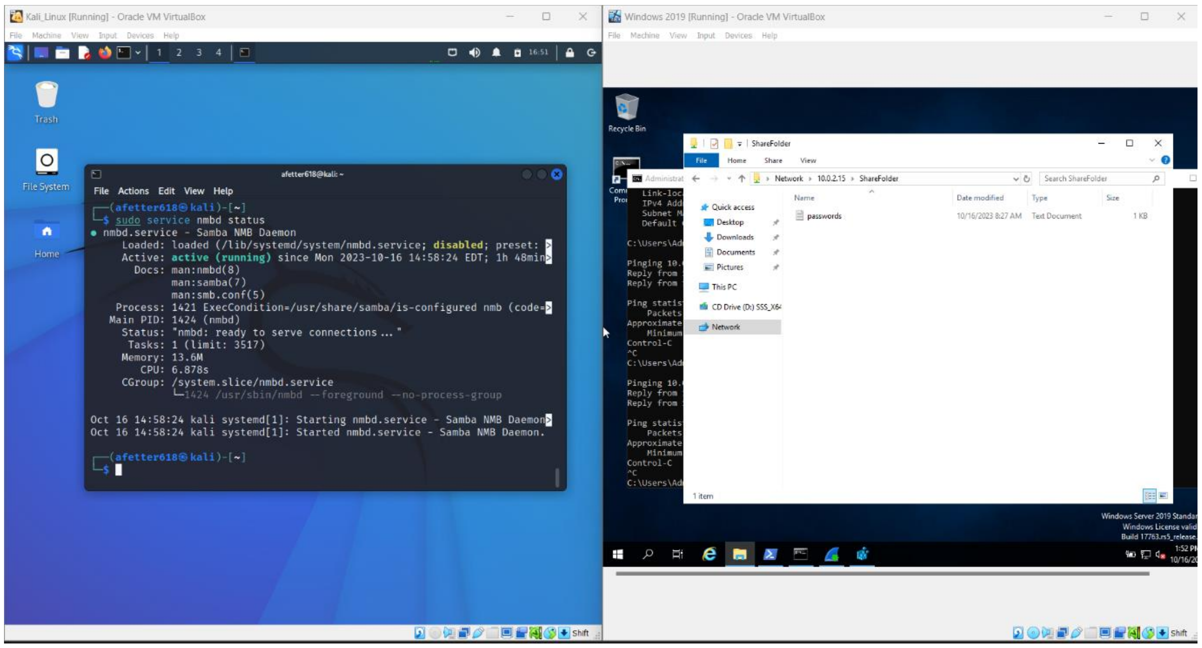
1. Press the Windows Key + R in the Windows VM and type “regedit” to open up the Registry Editor. This will allow you to manage user and group access to specific system registries.

4. Follow the file path “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters”, double click on “Parameters”, and change the value from 0 to 1

5. Open up the Run application once more and type “\\IP\share”, where “IP” is the IP address of the Linux machine and “Share” is the shared folder you created earlier.

6. You should now see the shared files on your Windows machine! Enter the credentials you made earlier to gain access to the files you created.

Example output:



```
┌──(afetter618㉿kali)-[~]
└─$ sudo service nmbd status
● nmbd.service - Samba NMB Daemon
     Loaded: loaded (/lib/systemd/system/nmbd.service; disabled; preset: >
     Active: active (running) since Mon 2023-10-16 14:58:24 EDT; 1h 48min>
       Docs: man:nmbd(8)
             man:samba(7)
             man:smb.conf(5)
    Process: 1421 ExecCondition=/usr/share/samba/is-configured nmb (code=>
   Main PID: 1424 (nmbd)
     Status: "nmbd: ready to serve connections..."
      Tasks: 1 (limit: 3517)
     Memory: 13.6M
        CPU: 6.878s
     CGroup: /system.slice/nmbd.service
             └─1424 /usr/sbin/nmbd --foreground --no-process-group

Oct 16 14:58:24 kali systemd[1]: Starting nmbd.service - Samba NMB Daemon>
Oct 16 14:58:24 kali systemd[1]: Started nmbd.service - Samba NMB Daemon.

┌──(afetter618㉿kali)-[~]
└─$ 
```

Kali_Linux [Running] - Oracle VM VirtualBox
Windows 2019 [Running] - Oracle VM VirtualBox
ShareFolder
Network › 10.0.2.15 › ShareFolder
Name | Date modified | Type | Size
passwords | 10/16/2023 8:27 AM | Text Document | 1 KB
1 item

5. Securing Samba

5.1. Team Split-Up

For this section, split members into three teams. Each team will have ~20-30 minutes to implement what is shown below. Only the tasks are shown when presenting and the answers are shown once the time is up. This allows the team to work together quickly and efficiently to achieve the task at hand in preparation for the real event. Remember, the competition is meant to be stressful!

5.1.1. Team 1

1. Enable host-based access control to the shared files with the following parameters:
    a. Allow your loopback address and VM address (with subnet) to access the content
    b. Deny all other addresses
2. Enable user-based access control to allow for only your user to access the shared files. This should be the username you used to access the files last week.
3. Enable guest-based access control by allowing guests to access the files but deny allowing **only** guests to access them. The guest accounts should be created using ftp.

5.1.2. Team 2

1. Enable interface protection and allow all ethernet interfaces and the loopback interface to listen for a connection from Samba
2. Require strong authentication to enable signatures and encryption when sending traffic while the Samba service is running

5.1.3. Team 3

1. Verify TLS peer certificates and names to enable privacy and data security for communication between hosts. Ensure that all checks are performed (verify the hosts digital certificate, hostname, IP address, etc.).
2. If you have extra time, write out a description explaining what each option does.

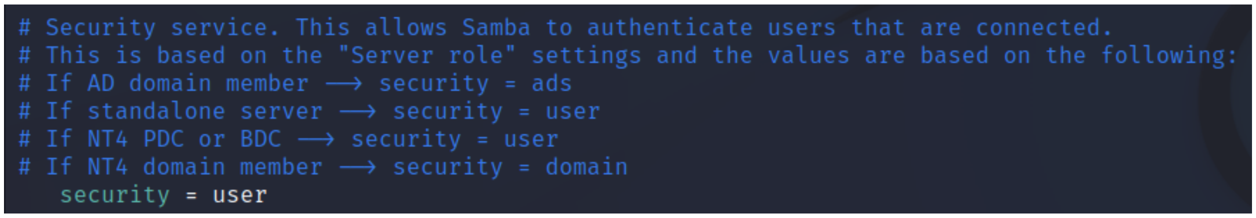
5.2. Answers to Team Exercise

5.2.1. Add the security service

The "security" service allows Samba to authenticate users accessing the contents. The values for this setting relies on the value of the "server role" setting found in the "Authentication" section of the /etc/Samba/smb.conf. For this service, the options are as followed:

1. If AD domain member, security = ads
2. If standalone server, security = user
3. If NT4 PDC or BDC, security = user
4. If NT4 domain member, security = domain

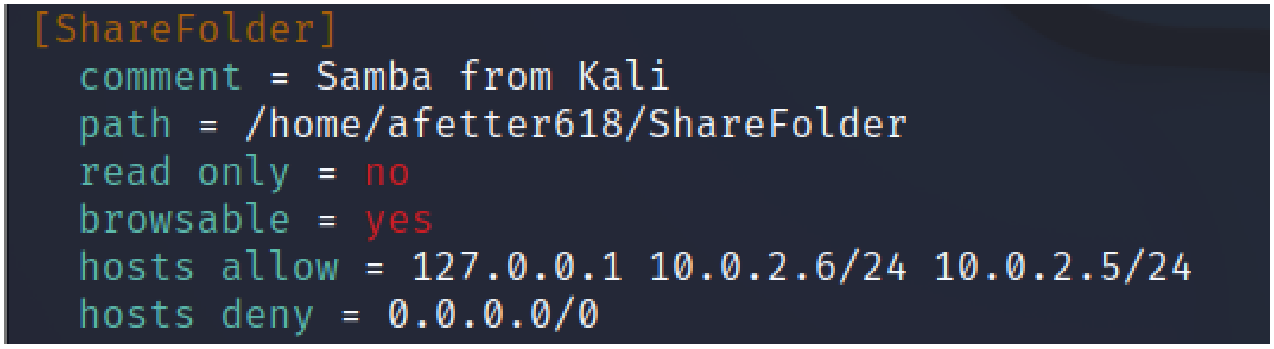
- Note that "security = user" used to be "security = share". So if you see it in any documentation, this is what it's referring to



```
# Security service. This allows Samba to authenticate users that are connected.
# This is based on the "Server role" settings and the values are based on the following:
# If AD domain member ⟶ security = ads
# If standalone server ⟶ security = user
# If NT4 PDC or BDC ⟶ security = user
# If NT4 domain member ⟶ security = domain
   security = user
```

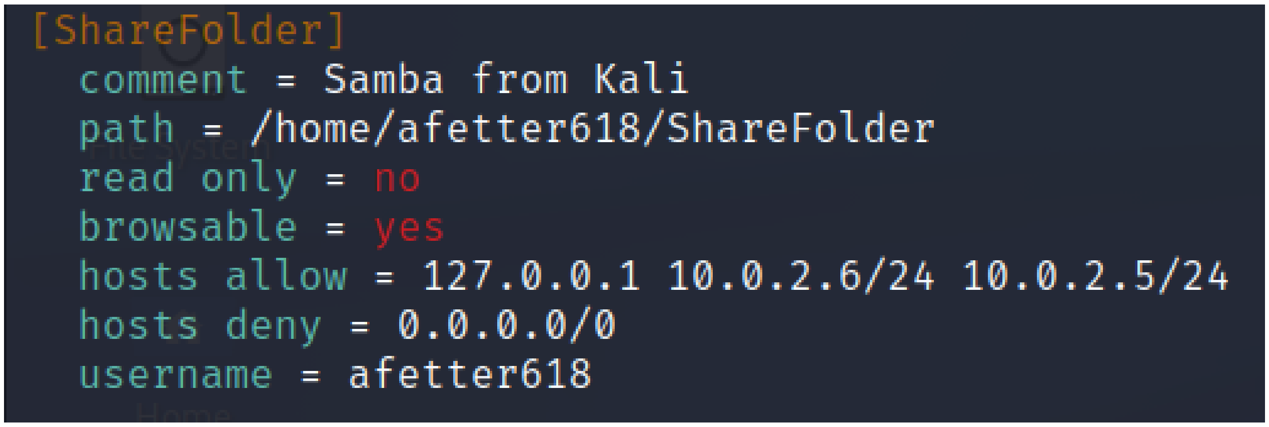
5.2.2. Enabling host-based, user-based, and guest-based protection

- By default, any host on the network can access the Samba server if they have the correct credentials. In order to limit the hosts that are allowed access (even if they have the credentials to get in), you can do the following:
  - In /etc/Samba/smb.conf, where you have implemented your share files, put the tags "hosts allow = IP" and "hosts deny = IP". This allows you to restrict access to only a particular host or set of hosts (see example below).
  - You can specify subnets by using "/#", "255.255.255.0", or simply "host."
  - You can also limit hosts on the subnet by using "IP/subnet except IP"



```
[ShareFolder]
  comment = Samba from Kali
  path = /home/afetter618/ShareFolder
  read only = no
  browsable = yes
  hosts allow = 127.0.0.1 10.0.2.6/24 10.0.2.5/24
  hosts deny = 0.0.0.0/0
```

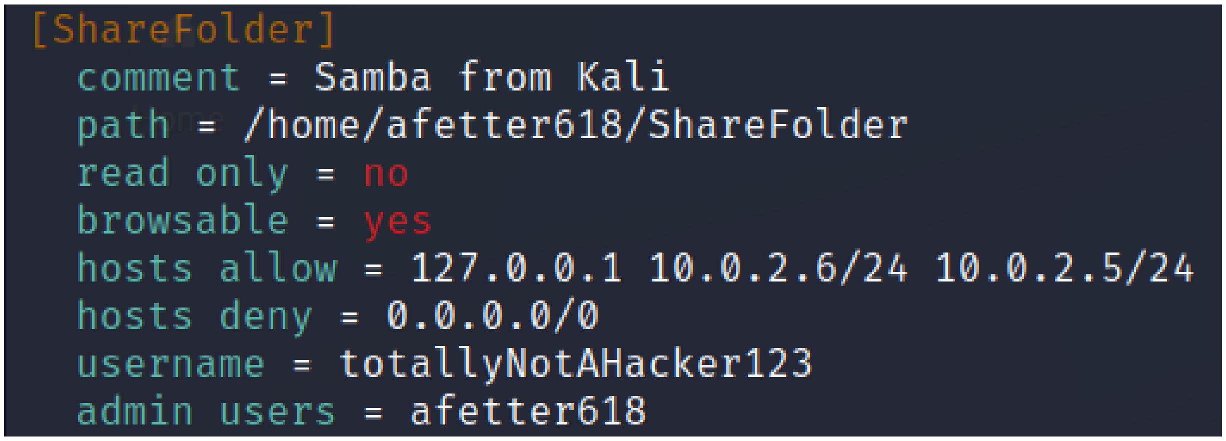
- Furthermore, you can limit user access by adding a similar field to the configuration file: "username: name"



```
[ShareFolder]
  comment = Samba from Kali
  path = /home/afetter618/ShareFolder
  read only = no
  browsable = yes
  hosts allow = 127.0.0.1 10.0.2.6/24 10.0.2.5/24
  hosts deny = 0.0.0.0/0
  username = afetter618
```

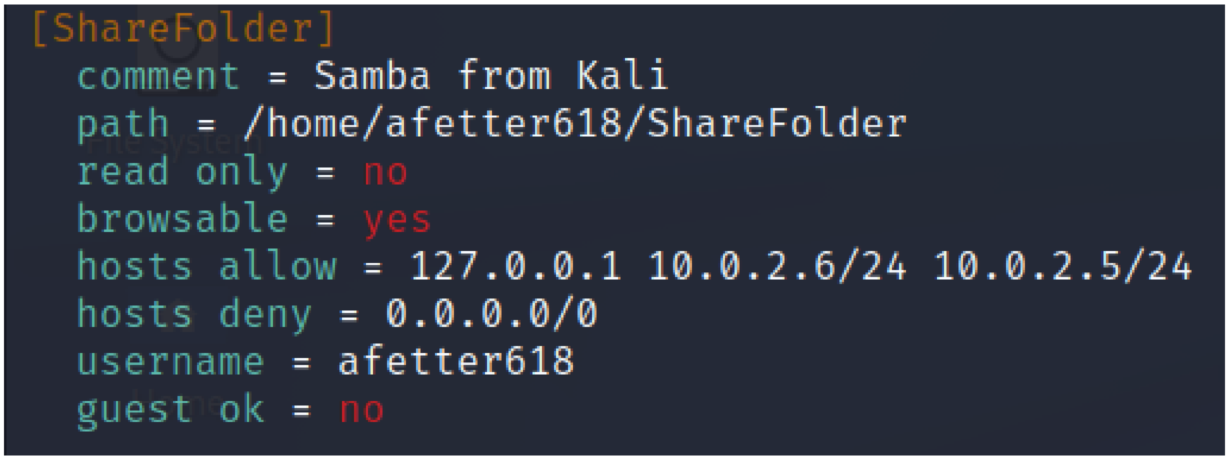
- To tell Samba which specific users have root privileges, you can use the "admin users = user" field

5. Securing Samba



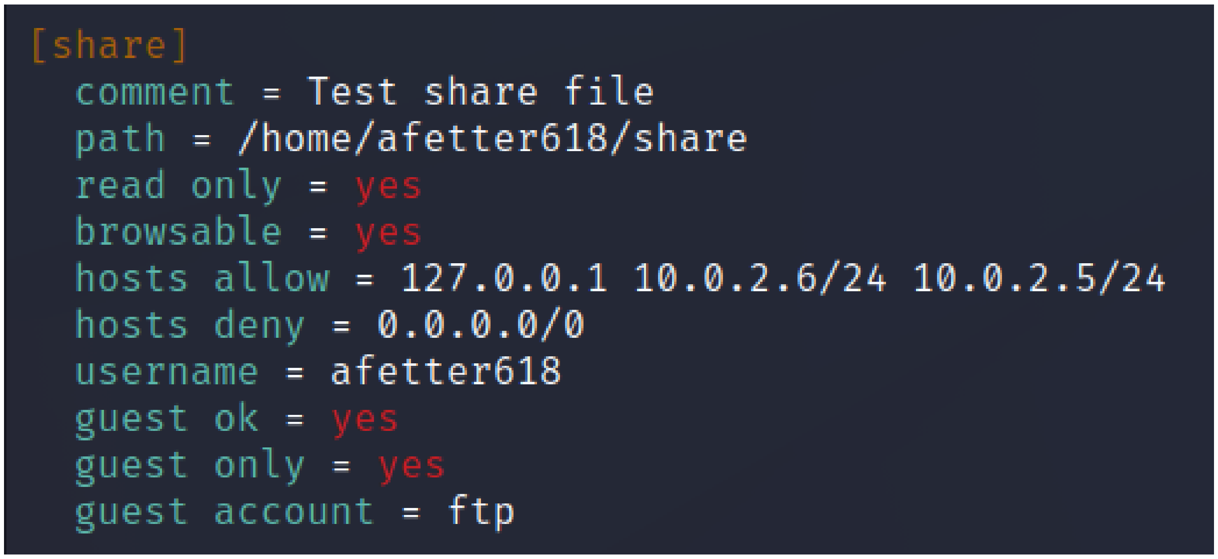
```
[ShareFolder]
   comment = Samba from Kali
   path = /home/afetter618/ShareFolder
   read only = no
   browsable = yes
   hosts allow = 127.0.0.1 10.0.2.6/24 10.0.2.5/24
   hosts deny = 0.0.0.0/0
   username = totallyNotAHacker123
   admin users = afetter618
```

- Finally, you can limit guest access by adding the field “guest ok = yes” or “guest ok = no” to the configuration file



```
[ShareFolder]
   comment = Samba from Kali
   path = /home/afetter618/ShareFolder
   read only = no
   browsable = yes
   hosts allow = 127.0.0.1 10.0.2.6/24 10.0.2.5/24
   hosts deny = 0.0.0.0/0
   username = afetter618
   guest ok = no
```

- If you want to enable guest access only without any user account, you can add the “guest only = yes” or “guest only = no” in conjunction with the “guest ok” field. You can also specify how you want guests accounts to be created using the “guest account” field

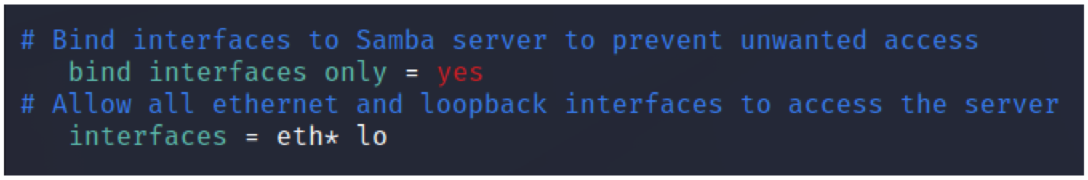


```
[share]
   comment = Test share file
   path = /home/afetter618/share
   read only = yes
   browsable = yes
   hosts allow = 127.0.0.1 10.0.2.6/24 10.0.2.5/24
   hosts deny = 0.0.0.0/0
   username = afetter618
   guest ok = yes
   guest only = yes
   guest account = ftp
```

*** There are several other fields that can be added to shares in order to make Samba more secure. However, these are the ones I wanted to highlight the most. You can find more information about the available fields here. ***

5.2.3. Enable interface protection

In order to enable interface protection and specify the interfaces that are allowed to connect to the Samba server, locate the /etc/Samba/smb.conf file and add the following in the "Global Settings" section



```
# Bind interfaces to Samba server to prevent unwanted access
   bind interfaces only = yes
# Allow all ethernet and loopback interfaces to access the server
   interfaces = eth* lo
```

- The "bind interfaces only" line tells the Samba server that only the specific interfaces listed are allowed to access it. If no interfaces are defined, no one will have access.
- Note that "etho*" and "lo" are specified in the "interfaces" line. This means that any ethernet interface (eth0, eth1, etc.) as well as the loopback interface will be listening for a connection from the Samba server and any other interface will not be able to connect.

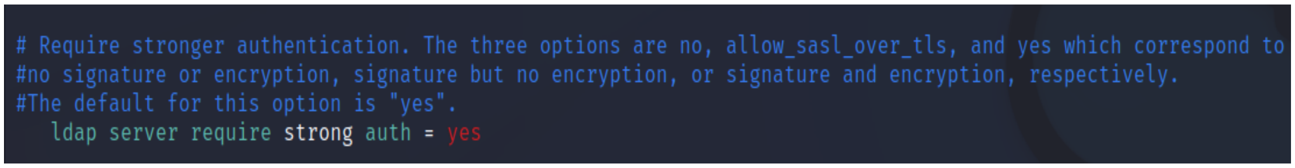
5.2.4. Require strong authentication

Samba uses a Lightweight Directory Access Protocol (LDAP) server to send data to other hosts. The "ldap server require strong auth" setting allows you to specify how you want traffic to be sent when the Samba service is running. You can add it by placing it in /etc/Samba/smb.conf "Global Settings" section.

<u>There are three options for this setting:</u>

1. no – No LDAP traffic signing or encryption
2. allow_sasl_over_tls – LDAP traffic signing but no encryption
3. yes – LDAP traffic signing AND encryption

By default, the setting is set to yes. However, this is not shown in the configuration file and should be added just in case.



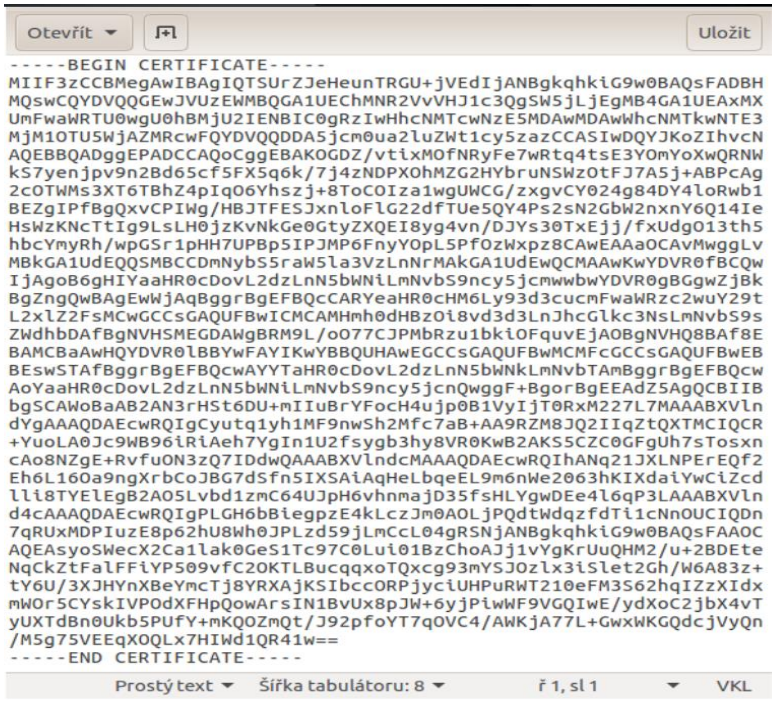
```
# Require stronger authentication. The three options are no, allow_sasl_over_tls, and yes which correspond to
#no signature or encryption, signature but no encryption, or signature and encryption, respectively.
#The default for this option is "yes".
   ldap server require strong auth = yes
```

5.2.5. *Verify TLS peer certificate and name*

Samba uses Transport Layer Security (TLS) to implement privacy and data security for communication between hosts. In order to ensure that a host is really who they say they are, the “tls verify peer” setting can be used in the /etc/Samba/smb.conf file.

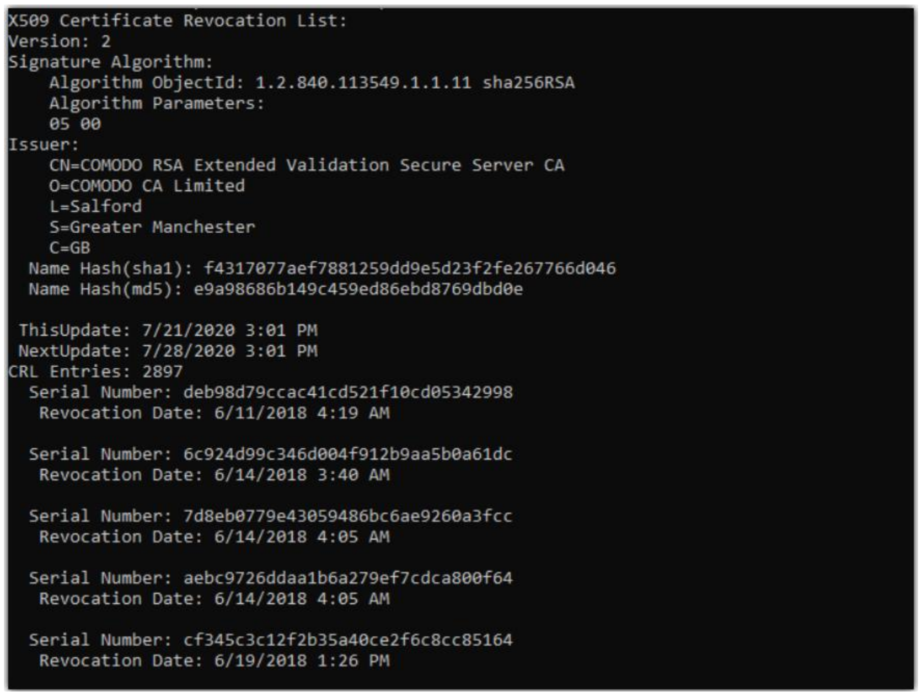
<u>The options for this setting include:</u>

1. no_check – There is no verification. This is a bad decision because it can easily induce man-in-the-middle attacks
2. ca_only – Only the certificate is verified. There must be another setting within the configuration file, “tls ca file” (ca stands for “Certificate Authority), that contains a valid certificate file is needed if this option is used. The “tls crl file” (optional) can also be added for extra security, but needs another file for the certificate revocation list (CRL)
   - Example of tls ca file:™



```
-----BEGIN CERTIFICATE-----
MIIF3zCCBMegAwIBAgIQTSUrZJeHeunTRGU+jVEdIjANBgkqhkiG9w0BAQsFADBH
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEgMB4GA1UEAxMX
UmFwaWRTU0wgU0hBMjU2IENBIC0gRzIwHhcNMTcwNzE5MDAwMDAwWhcNMTkwNTE3
MjM1OTU5WjAZMRcwFQYDVQQDDA5jcm0ua2luZWt1cy5zazCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAKOGDZ/vtixMOfNRyFe7wRtq4tsE3YOmYoXwQRNW
kS7yenjpv9n2Bd65cf5FX5q6k/7j4zNDPXOhMZG2HYbruNSWzOtFJ7A5j+ABPcAg
2cOTWMs3XT6TBhZ4pIqO6Yhszj+8ToCOIza1wgUWCG/zxgvCY024g84DY4loRwb1
BEZgIPfBgQxvCPIWg/HBJTFESJxnloFlG22dfTUe5QY4Ps2sN2GbW2nxnY6Q14Ie
HsWzKNcTtIg9LsLH0jzKvNkGe0GtyZXQEI8yg4vn/DJYs30TxEjj/fxUdgO13th5
hbcYmyRh/wpGSr1pHH7UPBp5IPJMP6FnyYOpL5PfOzWxpz8CAwEAAaOCAvMwggLv
MBkGA1UdEQQSMBCCDmNybS5raW5la3VzLnNrMAkGA1UdEwQCMAAwKwYDVR0fBCQw
IjAgoB6gHIYaaHR0cDovL2dzLnN5bWNiLmNvbS9ncy5jcmwwbwYDVR0gBGgwZjBk
BgZngQwBAgEwWjAqBggrBgEFBQcCARYeaHR0cHM6Ly93d3cucmFwaWRzc2wuY29t
L2xlZ2FsMCwGCCsGAQUFBwICMCAMHmh0dHBzOi8vd3d3LnJhcGlkc3NsLmNvbS9s
ZWdhbDAfBgNVHSMEGDAWgBRM9L/oO77CJPMbRzu1bkiOFquvEjAOBgNVHQ8BAf8E
BAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMFcGCCsGAQUFBwEB
BEswSTAfBggrBgEFBQcwAYYTaHR0cDovL2dzLnN5bWNkLmNvbTAmBggrBgEFBQcw
AoYaaHR0cDovL2dzLnN5bWNiLmNvbS9ncy5jcnQwggF+BgorBgEEAdZ5AgQCBIIB
bgSCAWoBaAB2AN3rHSt6DU+mIIuBrYFocH4ujp0B1VyIjT0RxM227L7MAAABXVln
dYgAAAQDAEcwRQIgCyutq1yh1MF9nwSh2Mfc7aB+AA9RZM8JQ2IIqZtQXTMCIQCR
+YuoLA0Jc9WB96iRiAeh7YgIn1U2fsygb3hy8VR0KwB2AKS5CZC0GFgUh7sTosxn
cAo8NZgE+RvfuON3zQ7IDdwQAAABXVlndcMAAAQDAEcwRQIhANq21JXLNPErEQf2
Eh6L16Oa9ngXrbCoJBG7dSfn5IXSAiAqHeLbqeEL9m6nWe2063hKIXdaiYwCiZcd
lli8TYElEgB2AO5Lvbd1zmC64UJpH6vhnmajD35fsHLYgwDEe4l6qP3LAAABXVln
d4cAAAQDAEcwRQIgPLGH6bBiegpzE4kLczJm0AOLjPQdtWdqzfdTi1cNnOUCIQDn
7qRUxMDPIuzE8p62hU8Wh0JPLzd59jLmCcL04gRSNjANBgkqhkiG9w0BAQsFAAOC
AQEAsyoSWecX2Ca1lak0GeS1Tc97C0Lui01BzChoAJj1vYgKrUuQHM2/u+2BDEte
NqCkZtFalFFiYP509vfC2OKTLBucqqxoTQxcg93mYSJOzlx3iSlet2Gh/W6A83z+
tY6U/3XJHYnXBeYmcTj8YRXAjKSIbccORPjyciUHPuRWT210eFM3S62hqIZzXIdx
mWOr5CYskIVPOdXFHpQowArsIN1BvUx8pJW+6yjPiwWF9VGQIwE/ydXoC2jbX4vT
yUXTdBn0Ukb5PUfY+mKQOZmQt/J92pfoYT7qOVC4/AWKjA77L+GwxWKGQdcjVyQn
/M5g75VEEqXOQLx7HIWd1QR41w==
-----END CERTIFICATE-----
```

3. ca_and_name_if_available – The above checks are performed as well as a verification of the user machine's hostname. The hostname is compared to the certificate's name provided in the "tls ca file" setting. If there is no hostname, the IP address will be compared



```
X509 Certificate Revocation List:
Version: 2
Signature Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.11 sha256RSA
    Algorithm Parameters:
    05 00
Issuer:
    CN=COMODO RSA Extended Validation Secure Server CA
    O=COMODO CA Limited
    L=Salford
    S=Greater Manchester
    C=GB
  Name Hash(sha1): f4317077aef7881259dd9e5d23f2fe267766d046
  Name Hash(md5): e9a98686b149c459ed86ebd8769dbd0e

 ThisUpdate: 7/21/2020 3:01 PM
 NextUpdate: 7/28/2020 3:01 PM
CRL Entries: 2897
  Serial Number: deb98d79ccac41cd521f10cd05342998
   Revocation Date: 6/11/2018 4:19 AM

  Serial Number: 6c924d99c346d004f912b9aa5b0a61dc
   Revocation Date: 6/14/2018 3:40 AM

  Serial Number: 7d8eb0779e43059486bc6ae9260a3fcc
   Revocation Date: 6/14/2018 4:05 AM

  Serial Number: aebc9726ddaa1b6a279ef7cdca800f64
   Revocation Date: 6/14/2018 4:05 AM

  Serial Number: cf345c3c12f2b35a40ce2f6c8cc85164
   Revocation Date: 6/19/2018 1:26 PM
```

4. ca_and_name – Performs the same checks as the "ca_and_name_if_available" option but compares both the hostname and the IP address to the certificate's name (not one or the other)
5. as_strict_as_possible – All checks mentioned above are performed. However, the "tls crl file" must be configured and is not optional.

By default, the setting is set to as_strict_as_possible. However, this is not shown in the configuration file and should be added just in case.

```
#Verify TLS peer certificate and name
   tls verify peer = as_strict_as_possible
```

6. Resources

- Official Samba Website
  - smb.conf File Documentation
- Oracle – Samba Documentation
- Ubuntu – Installing and Configuring Samba
- Pheonixnap - Samba Configuration
- Samba Wiki

6. Resources

  - Setting up Samba
  - Security Documentation
  - Samba 4.4 Added Features Added/Changed
- Red Hat Customer Portal – Using Samba as a Server
- Oreilly – Samba Security
- Linux-training.be - Samba Chapters