Securing a Samba-Fileserver inside an Active Directory

Author:
Stefan Kania

Ort:
SambaXP 2018 Göttingen

4. Juni 2018

Content

1 Introduction

In this years tutorial we will talk about securing a Samba-filserver inside an Active Directory-infrastructure. It doesn't matter whether you use a Microsoft-AD or a Samba4-AD, securing a Samba-fileserver will always follow the same steps. Here we will use a Samba4-domaincontroller.

2 Considerations

Always remember that you not just have to take a look at the Samba-security of a system, but you must also be aware of the system-security of your machine. There are ways to attack your system and your samba-service.

So let's take a first look at the thing you have to think about:

- Which distribution to use
  You should always use the distribution you already using in your company. Try not to build a distribution-zoo with lots of different distributions. If you use different distributions maybe the way you update your system is different and you may have different versions of the same service.
- Partition your system
  Never ever choose the default partition-setting of a distribution, always create your partitions by hand. As an example you can create the following partitions:
  - /
  - /boot (ro)
  - /var (noexec)
  - /usr (ro)
  - /tmp (noexec)
  - /data (rw,noexec)
  - /home

  The partition named */data* in the example will be used later to create the shares for the user-data.
- Choosing the version and the way to install Samba
  You should always look for the functions you need and then choose the Samba-version and the way how to install it. To install the Samba-version you can choose on of the following ways:
  - From source
    If you install Samba from the source you can have always the newest version of Samba running on your system. But the newest version is not always the best version. Remember when you install Samba from the sources you have to take care of the dependencies, the updates, and the security-patches on your own. If you are not building your own packages you will always have a build-environment on your server.

- Packages from the distribution
  If you choose the packages from the distribution, you will get all the update automatically, but you will never have the newest version of Samba. If you stay with a distribution for a long time, then maybe the samba-packages will not be supported by the samba-team, so you must rely on the maintainer of the distribution.
- The SerNet-packages
  The SerNet-packages are normally always up to date and it is easy to update to a newer Samba-Version, because you just have to change to a different repository for the packages, but you have to pay for the support of the packages.

3 System security

The fist step should always be to secure your system. Do all the following steps before you install and configure Samba.

3.1 Partitioning the system

As you can see the Linux-system is already installed and all the partitions are created, the only thing you have to do is changing the mount-options for the filesystems. Listing 3.1.1 will show you the options you have to set:

```
UUID=9b407b04-b645-4189-... / ext4 errors=remount-ro 0 1
UUID=4bd9fc5e-aa26-4d42-... /boot ext4 rw,exec 0 2
UUID=97d39b1c-e684-4ac9-... /data ext4 rw,noexec 0 2
UUID=6f39688a-4ab9-46db-... /home ext4 defaults 0 2
UUID=9d394f02-85b9-43e7-... /tmp ext4 defaults 0 2
UUID=daf4756d-ad97-40bc-... /usr ext4 ro,exec 0 2
UUID=fb1fbfd5-e235-4a0a-... /var ext4 defaults 0 2
UUID=9cf4566c-c868-4f0f-... none swap sw 0 0
```

Listing 3.1.1: Mount-options for /etc/fstab

After you set all the options in /etc/fstab just reboot the system to mount all filesystems withe the new options.

3.2 Deactivate unwanted protocols

Don't install any network-protocol you don't need, or leave a network-protocol unconfigured if it's installed. Here we don't want to use IPv6 so it should be disabled and all services installed should be configured so that they are not listening for IPv6 connections.

3.3 Deactivating IPv6

Check your interface with `ip a` and you will see, that your networkinterface uses IPv6, even if you don't want IPv6 and you don't have IPv6 active in your network. You should not only deactivate IPv6 for your networking device but also for all services running on your system, like *ssh*. To figure out, which services are running on your system use either

the new `ss`-command or the old `netstat`-command. On most of the actual distribution you have to install `netstat`.

To deactivate the IPv6-protocol, you have to create a file named /etc/sysctl.d/01-disable-ipv6.conf. You see the content of the file in listing 3.3.1:

```
net.ipv6.conf.all.disable_ipv6 = 1
```

Listing 3.3.1: content of 01-disable-ipv6.conf

After you have created the file do a `sysctl -p /etc/sysctl.d/01-disable-ipv6.conf`. When you check the networkinterface again with `ip a`, you will see, that there is no IPv6 part anymore. You could also reboot the system.

In the next step we will disable IPv6 for *ssh*, because if you would look again with `ss` or `netstat` you will see, that *ssh* is still listening to IPv6. To disable IPv6 open /etc/ssh/sshd_config with an editor and replace the line *AddressFamily any* with *AddressFamily inet*. After you have restarted ssh you will see, that ssh is not listening to IPv6 anymore.

4 Securing samba

We already installed all the needed packages for this tutorial, so you don't have to install any of the samba-packages and we can start directly with the configuration of samba.

The first thing, never use the /etc/samba/smb.conf which is installed together with the packages. This file is total useless for running a secure samba-fileserver. So remove the file now.

4.1 configuring Samba

After you have removed the file, we start with the configuration of the *[global]*-section of the /etc/samba/smb.conf. You can see all new parameters in listing 4.1.1:

```
[global]
        workgroup = example
        realm = EXAMPLE.NET
        security = ADS
        winbind use default domain = yes
        winbind refresh tickets = Yes
        template shell = /bin/bash
        idmap config * : range = 10000 - 19999
        idmap config EXAMPLE : backend = rid
        idmap config EXAMPLE : range = 1000000 - 1999999
        inherit acls = Yes
        store dos attributes = Yes
        vfs objects = acl_xattr
        interfaces = 192.168.56.101
        bind interfaces only = yes
        client ipc min protocol = smb2_10
        client min protocol = smb2_10
        disable netbios = yes
```

Listing 4.1.1: The new global-section

Let's explain all the parameters:

- workgroup = example
  Then name of the Windows-domain.
- realm = EXAMPLE.NET
  The Kerberos-Realm.
- security = ads
  Makes the system a domainmember.
- winbind use default domain = yes
  To see just the username without the domain when listing users or giving permissions.
- winbind refresh tickets = Yes
  Automatically refreshes the Kerberos-tickets.
- template shell = /bin/bash
  The default shell for AD-users when they login to the system.
- idmap config * : range = 10000 - 19999
  The UID-range for the build in users.
- idmap config EXAMPLE : backend = rid
  The backend to generate the UIDs of the AD-users.
- idmap config EXAMPLE : range = 1000000 - 1999999
  The UID-range for the AD-users.
- inherit acls = Yes
  Is needed to inherit the ACLs inside the filesystem when permissions are set via Windows.
- vfs objects = acl_xattr
  Manage filesystempermission the Windows-way.
- store dos attributes = Yes
  Is also needed for the filesystempermission.
- interfaces = 192.168.56.101
  If your system has more then one networkinterface, Samba will only bind to this interface. Also the localhost-address is note used. So for security-reasons it's always good to set this parameter.
- bind interfaces only = yes
  Only if you set this parameter to *yes* the *interfaces*-parameter will work.
- client ipc min protocol = smb2_10
  This will protect your system against connection over old unsecure ipc-protocol versions. A Windows XP system can not connect anymore.
- client min protocol = smb2_10
  This will protect your system against connection over old smb-protocol versions. A Windows XP system can not connect anymore.
- disable netbios = yes
  The System will not be seen anymore inside the networkneighborhood. But it may be not possible to join the system to a domain. After you have joined the domain yo can activate this parameter.

4.2 Configuring Kerberos

After you have configured Samba, you must configure the Kerberos-client before you can join to a domain. The Kerberos-client is configured via the file /etc/krb5.conf.You can see the settings in listing 4.2.1:

```
[libdefaults]
       default_realm = EXAMPLE.NET
       dns_lookup_realm = false
       dns_lookup_kdc = true
```

Listing 4.2.1: Settings for the kerberos-client

Remember, that the Kerberos-client will use DNS to find the Kerberos-Server. So you must have set the ADDC as your DNS-Server. Test if you can resolve the name of your ADDC, then you can try to get a Kerberos-ticket with `kinit administrator` and your administrator-password. Only if this works, you can join the domain.

4.3 Joining the domain

After you have setting up Samba an Kerberos AND you have tested DNS and Kerberos successfully, you can join the domain as shown in listing 4.3.1:

```
root@fs-01# net ads join -U administrator
  Enter administrator's password:
  Using short domain name -- EXAMPLE
  Joined 'FS01' to dns domain 'example.net'

root@fs-01# net ads testjoin
  Join is OK
```

Listing 4.3.1: Joining the domain

If you see the *Join is OK*-message then you can continue.

The next step will be to get all the AD-users and AD-groups into your system to set permissions. For this you must edit the file /etc/nsswitch.conf as you can see in listing 4.3.2:

```
passwd: compat winbind
shadow: files
group: compat winbind
```

Listing 4.3.2: Change in nsswitch.conf

Now you can test if you can get all users and groups from the domaincontroller with `wbinfo -u`, `wbinfo -g` and `getent passwd administrator`. If you see all users and the passwd-output from administrator, then everything is working.

4.4 Testing the Server

Now you can test the Server with `smbclient`. If you use `smbclient` the usual way with smbclient -L fs-01.example.net you will see an error as in listing 4.4.1:

```
root@fs-01# smbclient -L fs01.example.net
  Enter root's password:
  protocol negotiation failed: NT_STATUS_INVALID_PARAMETER_MIX
```

Listing 4.4.1: First test with smbclient

This is because smbclient uses SMB-version 1.0, but you allow only SMB-Version 2.1 and higher. If you change the protocol-version it will work, as you can see in listing 4.4.2:

```
root@fs-01# smbclient -L fs-01.example.net -m SMB3
  Enter root's password:
  Anonymous login successful

          Sharename Type Comment
          --------- ---- -------
          IPC$ IPC IPC Service

  Anonymous login successful

          Server Comment
          --------- -------

          Workgroup Master
          --------- -------
```

Listing 4.4.2: Second test with SMB-version 3.x

There will be no *Workgroup Master* listed, because we don't use *NetBIOS* anymore and so there will be no *Workgroup Master*.

The next test will show you which ports are used to provide the Samba-service. Listing 4.4.3 will show the output from `netstat -tlpn`. You can also use `ss -tlpn`:

```
root@fs-01# netstat -tlpn
  Aktive Internetverbindungen (Nur Server)
  Prot RecvQ SendQ Local Address Address State PID/Prog.
  tcp 0 0 192.168.56.101:139 0.0.0.0:* LISTEN 2106/smbd
  tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 809/sshd
  tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 1366/master
  tcp 0 0 192.168.56.101:445 0.0.0.0:* LISTEN 2106/smbd
```

Listing 4.4.3: Open ports the first test

As you can see, Samba provides it's service still over port 139. This port is only used for *smb over NetBIOS*. We don't use *NetBIOS* anymore, so we can deactivate this port. You can also test the system from the outside. The next test will show the ports with `nmap` as you can see in listing 4.4.4:

```
stefan@external-host$ nmap 192.168.56.101
  Starting Nmap 6.40 ( http://nmap.org ) at 2018-07-04 17:45 CEST
  Nmap scan report for 192.168.56.101
  Host is up (0.00012s latency).
  Not shown: 997 closed ports
  PORT STATE SERVICE
  22/tcp open ssh
  139/tcp open netbios-ssn
  445/tcp open microsoft-ds

  Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Listing 4.4.4: Nmap to test the ports

Again, you see port 139 as active. Now you should close the port 139, so Samba will not listen to this port anymore. To disable this port just add the line from listing 4.4.5 to the global-section of your smb.conf:

```
smb ports = 445
```

Listing 4.4.5: Disable port 139

Restart the *smbd*-service and do both tests again, in listing 4.4.6 you can see the result:

```
root@fs-01# netstat -tlpn
  Aktive Internetverbindungen (Nur Server)
  Prot RecvQ SendQ Local Address Address State PID/Prog
  tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 809/sshd
  tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 1366/master
  tcp 0 0 192.168.56.101:445 0.0.0.0:* LISTEN 2194/smbd

  stefan@externer-host$ nmap 192.168.56.101
  Starting Nmap 6.40 ( http://nmap.org ) at 2018-07-04 18:04 CEST
  Nmap scan report for 192.168.56.101
  Host is up (0.00064s latency).
  Not shown: 998 closed ports
  PORT STATE SERVICE
  22/tcp open ssh
  445/tcp open microsoft-ds

  Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Listing 4.4.6: Test without port 139

But what will happen if you test the connectivity with the command `smbclient` without port 139? In listing 4.4.7 you will see the result:

```
root@fs-01# smbclient -L fs-01.example.net -m SMB3
  Enter root's password:
  Anonymous login successful

          Sharename Type Comment
          --------- ---- -------
          IPC$ IPC IPC Service
  Connection to fs-01.example.net failed \
     (Error NT_STATUS_CONNECTION_REFUSED)
  NetBIOS over TCP disabled -- no workgroup available
```

Listing 4.4.7: smbclient without port 139

The smbclient-command rely on the NetBIOS-protocol. You disabled NetBIOS so no connection with `smbclient` is possible anymore.

5 Managing shares

After you have done all the security settings on your server it is time that you create the shares, so that users can use your new server. I will show you a solution, that will be easy to administrate and will be using only Windows-permissions on all shares for your user. So it is easy for you to use the same way and the same permissions you would use on a Microsoft fileserver.

5.1 The administrative share

The first share we will create will be an administrative share. Only a member of the group *domain admins* can access this share. This will be the only share for which you must create the directory and set the permissions directly on your server. So the first things to do is to create the directory as you can see in listing 5.1.1:

```
root@fs-01# mkdir /data/admin-share
root@fs-01# chgrp "domain admins" /daten/admin-share
root@fs-01# chmod 775 /daten/admin-share/
```

Listing 5.1.1: Creating the directory for the admin-share

Now create the share inside the smb.conf-file, as you can see in listing 5.1.2:

```
[admin-share]
        path = /data/admin-share
        browsable = no
        read only = no
        administrative share = yes
```

Listing 5.1.2: The admin-share in smb.conf

The parameter *administrative share = yes* turns the share into an administrative share only members of the group *domain admins* can access. You can compare this with the drive-share *D$*, *E$* on a Windows-system.

Now connect to your share as *administrator* on your Windows-system and create a new folder and set the permission to a single group. After you have changed your settings, try to change to this directory. If the administrator is not member of the group you gave permission to, you will get a *permission denied*, that is the normal behavior on a Windows-system.

5.2 Creating a user-share

After you have created the administrative share, you now can create the first share for your users. Use the directory you just created under Windows to create the share. In listing 5.2.1 you will see the parameters for this share:

```
[department]
        path = /data/admin-share/department
        browsable = no
        read only = no
        hide unreadable = yes
```

Listing 5.2.1: The user-share

Now the users can connect to the share and they will only see all directories where they have at least the read-permission. But be careful with this option.

Tip !

If you have large filesystems with a lot of directories an subdirectories it can take some time to list the content of the directory structure. Test the access before you allow your users to access the share.

6 Using the audit-function

Sometimes it can be very useful to audit thing the users are doing on a filesystem on your server. For this reason there is the *vfs module full_audit* you can activate in your shares. Let's activate the vfs-module inside your share. In listing 6.1 you see the parameters you have to put in your configuration for your share:

```
[department]
        path = /data/admin-share/department
        browseable = no
        read only = no
        hide unreadable = yes
        vfs objects = full_audit
        full_audit:success = mkdir rmdir read pread write \
                             pwrite rename unlink connect
        full_audit:prefix = %u|%I|%m|%S
        full_audit:failure = none
        full_audit:facility = local5
        full_audit:priority = notice
```

Listing 6.1: Add the vfs-module full_audit

Now you should start `journalctl -f` on a console of your fileserver and then access the share with a user on your Windows-client and create, modify and delete some entries. You will see an output for any of the tasks.

Consideration !

You should look at the law in your country if it's allowed to log the tasks a user is doing on a fileserver.

You can't log the login of a user to your server. The user logs in to the domain on, so there is now login on a fileserver, thanks to Kerberos ;-).

7 firwall

If you want or must run a firewall on your server, remember that you must protect the operatingsystem and the Samba-service. I created a firewall-script which will protect your system and your service. As always: There are alway more then one solution for a firewall-script. That is just one ;-). In listing 7.1 you will see the firewall-script. The script is also located on your VM for the fileserver:

```
#!/bin/bash
# Policies
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT

# allow Loopback
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# allow three way handshake
# for statefull inspection
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# drop SYN packages
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
iptables -I INPUT -m conntrack --ctstate NEW \
               -p tcp ! --syn -j DROP

# dro fragment packages
iptables -A INPUT -f -j DROP

# drop XMAS packages
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

# drop all NULL packages
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

# drop spoof packages
for SPOOF in 224.0.0.0/4 240.0.0.0/5 240.0.0.0/5 0.0.0.0/8 \
                  239.255.255.0/24 255.255.255.255; do
    iptables -A INPUT -d ${SPOOF} -j DROP
    done
    for SPOOF in 10.0.0.0/8 169.254.0.0/16 172.16.0.0/12 \
                      127.0.0.0/8 192.168.0.0/24 224.0.0.0/4 \
                                  240.0.0.0/5 0.0.0.0/8 ; do
        iptables -A INPUT -s ${SPOOF} -j DROP
done

# simple spoofing protection
iptables -I INPUT -m conntrack --ctstate NEW,INVALID -p tcp \
               --tcp-flags SYN,ACK SYN,ACK -j REJECT \
                       --reject-with tcp-reset

# simple DDoS-protection
iptables -A INPUT -p tcp -m tcp --tcp-flags SYN,ACK,FIN,RST \
               RST -m limit --limit 1/s -j ACCEPT

# drop all invalid packages
iptables -A INPUT -m state --state INVALID -j DROP
iptables -A FORWARD -m state --state INVALID -j DROP
iptables -A OUTPUT -m state --state INVALID -j DROP

# simple portscanner protection
# portscann IP will be disabled for 24 Hours
# (3600 x 24 = 86400 Seconds)
iptables -A INPUT -m recent --name portscan --rcheck \
               --seconds 86400 -j DROP
iptables -A FORWARD -m recent --name portscan --rcheck \
               --seconds 86400 -j DROP

# Free IP after 24 hours
iptables -A INPUT -m recent --name portscan --remove
iptables -A FORWARD -m recent --name portscan --remove

# Accept ICMP
iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j LOG \
               --log-level debug --log-prefix "PING IP_TABLES:"
iptables -A INPUT -p icmp --icmp-type 11 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 12 -j ACCEPT
```

```
# Bruteforce-SSH protection
iptables -A INPUT -p tcp -m tcp --dport 22 -m state \
               --state NEW -m recent --set --name SSH --rsource
iptables -A INPUT -p tcp -m tcp --dport 22 -m recent \
               --rcheck --seconds 30 --hitcount 4 --rttl --name SSH \
                       --rsource -j REJECT --reject-with tcp-reset
iptables -A INPUT -p tcp -m tcp --dport 22 -m recent --rcheck \
               --seconds 30 --hitcount 3 --rttl --name SSH --rsource \
                       -j LOG --log-prefix "SSH brute force "
iptables -A INPUT -p tcp -m tcp --dport 22 -m recent --update \
               --seconds 30 --hitcount 3 --rttl --name SSH --rsource \
                       -j REJECT --reject-with tcp-reset
iptables -A INPUT -p tcp -m tcp --dport 22 -m state --state NEW \
               -m recent --update --seconds 600 --hitcount 3 --rttl \
                       --name SSH -j DROP

# Max 10 connections over Port 445 from each IP
iptables -A INPUT -p tcp -m tcp --syn --dport 445 -m connlimit \
              --connlimit-above 10 -j REJECT --reject-with tcp-reset

# Accept SMB
iptables -A INPUT -p tcp -m tcp --dport 445 -j ACCEPT

# Accept SSH
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT

# Accept Ping
iptables -A OUTPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
```

Listing 7.1: A firewall-script

I put a lot of gimmicks into the script maybe they are useful for you too.

Now you have a firewall-script to activate your firewall, the only thing you need now is a script for the *systemd* so the firewall will start, every time you reboot your system. In listing 7.2 you will see the script. You will find the script on your VM for the fileserver:

```
# File /etc/systemd/system/samba-firewall.service
[Unit]
Description=Samba-Firewall
After=syslog.target network.target

[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=/root/fire.bash

[Install]
WantedBy=multi-user.target
```

Listing 7.2: Firewall systemd-script

Copy the script to /etc/systemd/system, then activate the script with `systemctl enable samba-firewall.service` and start the script with `systemctl start samba-firewall`.

7.1 Test the firewall

After you have started the firewall you should test the firewall. *Namp* is a good tool to test the firewall. Nmap is not only a good tool to test your firewall but also to test your Samba-fileserver for security-gaps. For checking Samba you should always take a look alt `https://nmap.org/book/nse.html` to find out how you can use additional scripts with nmap to test your system. On `https://nmap.org/nsedoc/index.html` you will find a list of all supported scripts for nmap.

Let's take a look at some of the tests. In the following example I will use the script *samba-vuln-cve-2012-1182*, this is a test for an older Samba-Versions (¡3.6). This is a test you should use if you have an old NAS-box running an older Samba-version. If your system is vulnerable for this bug, a hacker could execute RPC-commands on your system without any login. In listing 7.1.1 you will see the output of this test:

```
root@meta# nmap --script=samba-vuln-cve-2012-1182 -p 139 192.168.56.101
  Starting Nmap 7.40 ( http://nmap.org ) at 2017-08-20 16:46
  Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing
    ARP Ping Scan
  ARP Ping Scan Timing: About 100.00% done; ETC: 16:46
  Nmap scan report for 192.168.56.101
  Host is up (0.00032s latency).
  PORT STATE SERVICE
  139/tcp closed netbios-ssn
  MAC Address: 08:00:27:AC:3A:85 (Cadmus Computer Systems)

  Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```

Listing 7.1.1: Test for cve-2012-1182

This system is save, if your system is not save, the output from nmap would look like the one in listing 7.1.2:

```
PORT STATE SERVICE
139/tcp open netbios-ssn

Host script results:
| samba-vuln-cve-2012-1182:
| VULNERABLE:
| SAMBA remote heap overflow
| State: VULNERABLE
| IDs: CVE:CVE-2012-1182
| Risk factor: HIGH CVSSv2: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C)
| Description:
| Samba versions 3.6.3 and all versions previous to this are affected by
| a vulnerability that allows remote code execution as the "root" user
| from an anonymous connection.
|
| Disclosure date: 2012-03-15
| References:
| http://www.samba.org/samba/security/CVE-2012-1182
|_ http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1182
```

Listing 7.1.2: unsave system for cve-2012-1182

As you can see, this bug effects only system with NetBIOS running. So if you have an old NAS-box and the box is vulnerable for this attack, it can help to deactivate NetBIOS as you did for your fileserver.

Let's check if your system is vulnerable against the *SambaCry*. You can do the test with the nmap-script *smb-vuln-cve-2017-7494* to test your system. In listing 7.1.3 you will see the test:

```
root@meta# nmap --script smb-vuln-cve-2017-7494 \
          --script-args smb-vuln-cve-2017-7494.\
                check-version -p445 192.168.56.101

  Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-20 18:39
  Nmap scan report for fs-01.example.net (192.168.56.101)
  Host is up (0.00067s latency).
  PORT STATE SERVICE
  445/tcp open microsoft-ds
  MAC Address: 08:00:27:AC:3A:85 (Oracle VirtualBox NIC)

  Host script results:
   smb-vuln-cve-2017-7494:
     VULNERABLE:
     SAMBA Remote Code Execution from Writable Share
       State: LIKELY VULNERABLE
       IDs: CVE:CVE-2017-7494
       Risk factor: HIGH CVSSv3: 7.5 (HIGH)
          (CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H)
       All versions of Samba from 3.5.0 onwards are vulnerable
       to a remote code execution vulnerability, allowing a
       malicious client to upload a shared library to a
       writable share, and then cause the server to load
       and execute it.

       Disclosure date: 2017-05-24
       Check results: Samba Version: 4.4.4
     References:
     https://www.samba.org/samba/security/CVE-2017-7494.html
     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-7494

  Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
```

Listing 7.1.3: Test for SambaCry

Here it gives you the result *State: LIKELY VULNERABLE*, because we are using a Samba-version which is vulnerable for this bug. You should then check, if the Samba-version you are using is already patched.

One last test you should do here in this tutorial is to check your firewall for a bruteforce-attack to the ssh-service. Be sure that your firewall is active then run the test as you see it in listing 7.1.4:

```
root@meta:~# nmap -p 22 --script ssh-brute --script-args \
                      userdb=users.lst,passdb=pass.lst \
                      --script-args ssh-brute.timeout=9s \
                      192.168.56.101

Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-22 19:59 CEST
Nmap scan report for fs-01.example.net (192.168.56.101)
Host is up (0.00048s latency).
PORT STATE SERVICE
22/tcp closed ssh
MAC Address: 08:00:27:AC:3A:85 (Oracle VirtualBox virtual NIC)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

Listing 7.1.4: Test for ssh bruteforce

As you can see, there are too files users.lst and pass.lst. Create this two files with some names and passwords to use with the test. As you can see, the result is that the ssh-port is closed, because your firewall detects the bruteforce-attack and closed the port for this IP-address.

Let's take a look at the log-file. See listing 7.1.5 for the result:

```
Jun 22 19:59:42 fs-01 kernel: SSH brute force IN=enp0s8OUT= \
MAC=08:00:27:ac:3a:85:08:00:27:a8:d5:34:08:00 \
SRC=192.168.56.167 DST=192.168.56.101 LEN=44 TOS=0x00 \
PREC=0x00 TTL=53 ID=56974 PROTO=TCP SPT=43282 DPT=22 \
WINDOW=1024 RES=0x00 SYN URGP=0
```

Listing 7.1.5: Logfile output

8 Conclusion

As you have seen in this years tutorial there are always a lot of things you have to do to run a save Samba-fileserver. Always remember that you have to secure the operatingsystem AND the Samba-Service.

Index