SAMBA – Brücke zwischen den Welten

Andreas Kretschmer mailto:andreas_kretschmer@despammed.com 18. Oktober 2003

Die Welt besteht nicht nur aus *NIX-Systemen, daher steht man als Admin oft genug vor der Aufgabe, auch PCs, die unter dem OS eines amerikanischen Herstellers von Spielekonsolen laufen, mit File- und Druckdiensten zu versorgen. Und ja, Pinguine vertragen auch Äpfel. Aber darauf gehe ich hier nicht ein. Ich möchte hier eine einfache Installation und Konfiguration eines SAMBA-Servers vorstellen, mit Windows2000- Clients, die in einer durch SAMBA kontrollierten Domäne sind. Zur Passwortverwaltung dient smbpasswd.

1 Was ist SAMBA?

1.1 Geschichte

In einer gemischten Umgebung, der der man untereinander kommunizieren will, müssen sich alle Teilnehmer auf den kleinsten gemeinsamen Nenner einigen.

Auch wenn Linux in solch einer Umgebung mit z.B. NFS oder anderen Netzwerkdiensten aufwarten kann - Windows-Rechner können es, zumindest von Haus aus, nicht. Was aber Microsoft all seinen neueren Produkten beilegt, ist das SMB-Protokoll.

Um also in einem Windows-Netzwerk mit seinem Pinguin nicht nur mitreden zu können, sondern standesgemäß den Ton angeben zu können, ist SAMBA das Mittel der Wahl.

Andrew Tridgell, Informatikstudent am Computer Science Lab der Australia National Univerity in Canberra, versuchte 1991 eine Verbindung zwischen seinem DOS-Rechner und dem SMB-Server von DCE aufzubauen.

Er begann, die Kommunikation zwischen Client und Server zu analysieren. Er schrieb ein Analysetool, das er *sockspy.c* nannte und das auch lange Zeit zu SAMBA gehörte. Irgendwann verstand er die Kommunikation und bastelte ein einfaches Serverprogramm. Im Januar 1992 veröffentlichte er `Server 0.1`, im monatlichen Abstand folgten `Server 0.5` und `Server 1.0`.

Nachdem er Linux entdeckte ;-), portierte er das nach Linux, es lief auf Anhieb. Er hatte inzwischen auch die Dokumentation zu dem SMB-Protokoll erhalten und begann, das Programm weiterzuentwickeln. Dennoch blieb die Entwicklung durch Reverse-Engineering kennzeichnend für SAMBA, insbesondere, weil Microsoft das Protokoll zwar ständig weiterentwickelte, aber keine Details veröffentlichte. Mittlerweile ist der 'harte Kern' der Entwickler ca. 20 Personen groß.

1.2 Details

Das SMB-Protokoll (Server Message Block) legt fest, wie Windows-Computer im Netzwerk kommunizieren. SMB ermöglicht den Zugriff auf Dateien, Drucker, serielle Schnittstellen und Kommunikationskanäle wie benannte Pipes und Mail-Slots. SAMBA ist eine freie SMB-Implementierung.

Der Ursprung von SMB ist gemeinsam mit NetBIOS, was ursprünglich sowohl Programmierschnittstelle als auch Transportprotokoll war und 1983 von Sytec für IBM programmiert wurde. Das Transportprotokoll nennt IBM NetBIOS Frame Protocol (NBF), was häufig auch als NetBIOS-Protokoll bezeichnet wird.

Im Laufe der Zeit zeigte sich, daß NetBIOS den Anforderungen größerer Netze nicht gerecht wurde, das NBF wurde durch das NetBIOS Extended User Protocol ersetzt.

Da es keinerlei Konfiguration erfordert (außer daß jeder Rechner einen eigenständigen NetBIOS-Namen haben muß), setzte es sich schnell in Windows-Netzwerken durch.

Später wurden Möglichkeiten geschaffen, SMB-Daten in anderen Transportprotokollen zu kapseln. Dies ermöglicht es, Netzwerkfreigaben auch über Routergrenzen hinweg zu nutzen. Die Kapselung in TCP/IP wird von MS NetBIOS over TCP/IP (NBT, NetBT) genannt.

Zentraler Punkt dieser Integration ist die Auflösung der NetBIOS-Namen in IP-Adressen. Sie erfolgt über Namens-Server für NetBIOS, den NBNS (NetBIOS Name Server), Microsoft verwendet dafür den Begriff WINS (Windows Internet Name Service).

Für diese ganze Geschichte, also der korrekten Namensauflösung, dient unter SAMBA der *nmbd*.

Für das Bereitstellen der Freigaben ist der *smbd* zuständig.

2 Installation

In der Regel verwendet man das der jeweiligen Distribution beiliegende Paket und installiert es mit den dafür vorgesehenen Tools.

Unter Debian also:

```
kretschmer@kaufbach:~$ apt-cache search samba | grep "^samba"
samba - a LanManager-like file and printer server for Unix
samba-common - Samba common files used by both the server and the client
samba-doc - Samba documentation
```

Für andere Distributionen sollte man halt die beiliegenden Tools zur Paketverwaltung nutzen.

2.1 Und selberbauen?

Der Eigenbau erfolgt, nach dem Download der Quellen, i.d.R. unkompliziert nach dem Linux-Dreisatz.

Also die Quellen runterladen, diese z.B. in /usr/src/ entpacken (tar -xvzf ...). Es entstehen einige Unterverzeichnisse, u.a. *docs*. Dieses sollte man sich anschauen, die dortigen Dateien sind oft hilfreich.

Nun erst einmal die Doku lesen, man kann mit Parametern für `configure` noch diverse Dinge einstellen und seinen Bedürfnissen anpassen.

Der zweite Schritt ist configure, dann erfolgt make, make install.

2.2 Auf die Plätze - fertig - los!

Der Start von SAMBA (smbd und nmbd) erfolgt üblicherweise über die Startscripte, auch hier liefern die Distributionen passende dazu. Dem Vernehmen nach startet bei SuSE 8.x defaultmäßig der nmbd nicht mit, zumindest klang das gelegentlich in Newsgroups an.

Die Folge ist, daß der SAMBA-Server nicht in der Netzwerkumgebung der Clients auftaucht. Wer also dieses Problem hat, sollte als erstes prüfen, ob der *nmbd* läuft.

3 Benutzer

Linux und auch Windows ab NT unterscheidet nach Benutzer, damit wird auch ein Benutzerkennwort erforderlich. Da SAMBA unter Unix läuft, nutzt es natürlich dessen Benutzerkonzept. Leider ist dieses Konzept vom WindowsNT-Konzept völlig verschieden. Die daraus resultierenden Probleme müssen gelöst werden.

Normalerweise sollte jedem Benutzer ein Benutzerkonto eingerichtet werden. Man kann allerdings auch einrichten, daß ein Benutzer unter anderem Namen am Client als am Unix angemeldet ist. Dazu kann man eine Mapping-Datei nutzen, darauf möchte ich hier aber nicht weiter eingehen, sondern nur auf die Möglichkeit verweisen.

3.1 Verschlüsselung

Frühere Versionen des SMB übertrugen Passwörter im Klartext, mittlerweile, mit Win 9x/NT, werden Passwörter verschlüsselt übertragen. SAMBA kann mit beiden Varianten umgehen.

Falls Sie ältere Clients haben, die nicht verschlüsseln können, müssen Sie SAMBA so einstellen, daß Klartext-Passwörter akzeptiert werden. Clients, die normalerweise verschlüsselt übertragen wollen, müssen dann auf unverschlüsselt umgestellt werden. Im SAMBA-Paket finden sich dazu passende Registry-Anweisungen.

Wo immer möglich sollte dies aber nicht erfolgen, sondern mit verschlüsselten Passworten gearbeitet werden.

Da Windows eine andere Verschlüsselungsroutine als Linux benutzt, sind die Linux-Passwortdateien (passwd, shadow) oder die NIS-Datenbank so nicht nutzbar. SAMBA verwendet die `smbpasswd` im `private`-Verzeichnis.

Zur Verwaltung der verschlüsselten SMB-Passwörter dient das Programm `smbpasswd`.

Wer von unverschlüsselten zu verschlüsselten Passwörtern migrieren will, kann mittels folgendem Aufruf erst einmal alle Benutzer aus `/etc/passwd` in der `/usr/local/samba/private/smbpasswd` erfassen:

```
cat /etc/passwd | /usr/src/samba/script/mksmbpasswd.sh > /usr/local/samba/private/smbpasswd
```

Damit sind alle Passwörter allerdings auf leer gesetzt, weil es keinen Weg gibt, ein Passwort zu entschlüsseln.

Ich habe das vor einiger Zeit auch an einigen Firmenstandorten gemacht, anschließend mit folgenden Einträgen in der `smb.cond` den Umstieg mir erleichtert:

```
update encrypted = yes
encrypt password = no
```

Nach einiger Zeit, wenn sich alle wenigstens einmal angemeldet haben, disabled man das update und enabled verschlüsselte Passwörter - die User merken davon nix.

3.2 Browsing

Unter Windows hat man die 'Netzwerkumgebung'. Sicher hat der eine oder andere schon einmal bemerkt, daß es da mitunter lustige Effekte gibt. Beispiel:

1. Rechner A ist zwar nicht zu sehen, aber wenn man seinen Namen kennt, kann man dennoch darauf zugreifen
2. Rechner B ist zu sehen, aber ein Zugriffsversuch schlägt fehl

Nun, die Funktion, die die Netzwerkumgebung aufbaut, nennt sich *Browsing* und zählt nicht direkt zu den Glanzleistungen im Netzwerkbereich.

3.2.1 Browsing - kurz erklärt

Um alle Rechner im Netz in der Netzwerkumgebung zu sehen, muß erst einmal eine Liste der Rechner aufgebaut werden.

Für jede Arbeitsgruppe wird ein Rechner zum 'Chef' erkoren, er ist der *Masterbrowser* Er sammelt alle IP-Adressen und Rechnernamen, die er finden kann, und erstellt diese Liste. Außerdem registriert er für sich den Namen `__MSBROWSER__`.

Das größte Problem an dieser Liste ist die Aktualität. Die Rechner melden sich beim booten an, außerdem regelmäßig im Betrieb. Diese Meldungen erfolgen als UDP-Broadcasts. Da UDP keine gesicherte Verbindung darstellt, kann der Masterbrowser nicht sofort einen Client aus seiner Liste streichen, wenn mal kein Paket kommt. Dies erfolgt erst nach 36 Minuten. Wir haben Fall B.

Da der Masterbrowser bestrebt ist, seinen Job sorgfältig zu machen, versucht er, seine Liste an einen Backup-Browser zu geben. Dadurch braucht er nicht selbst alle Anfragen von Clients nach der Liste beantworten, sondern kann es an den Backup-Server delegieren. Da dieser aber nur aller 15 Minuten aktualisiert wird, kann Fall 1 eintreten.

Nun stellt sich noch die Frage: Wer wird Masterbrowser?

Dazu hat MS seinen Betriebssystemen einen festen Wert mitgegeben:

```
WfW                          1
W9x                          2
WinNT Workstation 3.51      16
WinNT Workstation 4.0       17
WinNT Server 3.51           32
WinNT Server 4.0            33
```

Und SAMBA?

Das ist bei SAMBA der OS-Level und ist frei definierbar. Mit 64 gewinnt er immer ;-)

```
preferred master       Browser-Wahl bei jedem Start von nmbd durchführen
local master           an der Browser-Wahl teilnehmen
```

Damit kann man noch steuern, ob eine Wahl erzwungen werden soll und ob SAMBA als Masterbrowser überhaupt zur Verfügung stehen soll. Es kann nur einen geben!

4 Konfiguration

Die zentrale Konfiguration von SAMBA erfolgt in der Datei *smb.conf*, die distributionsabhängig an unterschiedlichen Orten sich befinden kann. Änderungen an dieser Datei bemerkt SAMBA selbständig, man braucht also nach Änderungen den Dienst i.d.R. nicht neustarten.

Der Aufbau dieser Datei ist simpel:

Es gibt einzelne Sektionen, die die für diese Sektion gültigen Optionen beinhaltet. Ich möchte dies an einem Beispiel erklären:

4.1 Server

4.1.1 Global

Zuerst einmal *[global]*

```
# Global parameters
[global]
        security = user
        encrypt passwords = Yes
        workgroup = SGS
        browsable = Yes
        browse list = Yes
        netbios name = pinguin
        interfaces = 192.168.1.1/255.255.255.0 lo
        bind interfaces only = yes
#
        read bmpx = Yes
        max xmit = 8192

# nach 10 Minuten werden Verbindungen ohne offene Files als geschlossen betrachtet
        deadtime = 10


        keepalive = 30
        read size = 8192
        socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192

# wichtig: unser Logon-Script. Relativ zu [netlogon]!
# Wichtig: DOS-Datei mit DOS-Zeilenenden!
        logon script = LOGON.BAT

# für uns wichtig
        domain logons = Yes

# damit gewinnt SAMBA gegen eNTen
        os level = 64

# JA!, wir sind der Master-Browser!
        preferred master = Yes
        domain master = Yes
        local master = Yes

# Ja!, wir sind auch WINS-Server
        wins support = Yes


        create mask = 0600

        character set = iso8859-1
        client code page = 850
```

```
        logon drive = y:


# angeblich braucht Access folgendes:
        oplocks = yes
        strict locking = yes

        printcap name = /etc/printcap
        load printers = yes
```

Damit ist der grundlegende Serverdienst konfiguriert - kommen wir nun an die eigentlichen Shares:

4.1.2 Logon und Rechner-Verzeichnis

Das Rechner-Verzeichnis ist eine Sache, die ich mir eingerichtet habe, dazu später mehr.

```
[netlogon]
        comment = Netzwerk-Login
        # preexec = echo \"%u connected to %S from %m (%I)\" >> /tmp/samba.log
        path = /usr/local/samba/netlogon

[machine]
        comment = Netzwerk-Login
        path = /usr/local/samba/netlogon/%m
```

Man kann seine Konfiguration mittels *testparm* aus dem SAMBA-Paket testen. Damit können Syntaxfehler leicht gefunden werden, natürlich keine logischen Fehler wie falsche Rechte.

4.1.3 Logon-Script

Über das Logon-Script, welches bei einer Domänen-Anmeldung auf dem Client ausgeführt wird, kann man auf den Clients diverse Dinge tun, u.a. Uhrzeit stellen und Laufwerke zu Shares mappen. Achtung: um Win2000-Clients mit der Uhrzeit versorgen zu können, muß man in den lokalen Sicherheitseinstellungen authentifizierten Benutzern das Recht zur Uhrzeiteinstellung einräumen.

```
net time \\pinguin /set /yes
net use X: /d
net use X: \\pinguin\machine
call X:\machine.bat
net use X: /d
net use W: \\pinguin\Arbeitsvorbereitung
net use X: \\pinguin\Allgemein
net use Y: \\pinguin\homes
call y:\user.bat
```

Der 2. Befehl unmountet ein evtl. vorhandenes gemapptes Share an X:, um dort das Share "machine" zu mappen. Dieses Share ist in der smb.conf als `/usr/local/samba/netlogon/%m` definiert, das Makro `%m` wird mit dem NetBIOS-Namen des Clients erweitert. Ich habe für einige Rechner spezielle Mappings, z.B. für Rechner, an denen ein Scanner angeschlossen ist. Daher rufe ich mit dem 4. Befehl dann eine rechnerspezifische Startdatei auf.

Weiter rufe ich eine Userspezifische Batch-Datei auf. Dies enthält dann z.B. für unterschiedliche Mitarbeiter unterschiedliche Mappings von Laufwerken. So erreiche ich, daß sich z.B. ein Mitarbeiter auch an einen Rechner eines anderen anmelden kann und dennoch seine gewohnten Laufwerke gemappt vorfindet.

Zu beachten ist bei diesen Startdateien, daß diese auf dem Client durchgeführt werden, also müssen sie dort korrekt lesbar sein. Daher Achtung, wenn man diese auf dem Server editiert, es müssen die DOS-typischen Zeilenumbrüche vorhanden sein.(vi: set ff=dos)

Wie im Beispiel gesehen, können diverse Makros verwendet werden. Hier einige dieser Makros:

```
%u   Username
%m   NetBios-Name des Clients
%I   IP-Adresse von Client
%d   PID des Serverprozesses
%a   Betriebssystemversion vom Client
%T   aktuelle Datum und Uhrzeit
```

4.1.4 HOME

```
[homes]
        #preexec = echo \"%u connected to %S from %m (%I)\" >> /tmp/samba.log
        comment = Heimatverzeichnis
        veto files = .*
        writeable = Yes
        browseable = No
# Achtung: folgendes ist wichtig für Win200 ab SP2 zur Ablage der Profile
        nt acl support = no
```

4.1.5 spezielle Anforderungen

Manchmal ist z.B. ein für alle frei zugängliches Verzeichnis recht nett. Ich nenne ein solches Share "Allgemein":

```
[Allgemein]
        comment = Allgemein
        path = /usr/local/samba/Allgemein
        read only = no
        public = yes
        guest ok = yes
        create mask = 0666
        directory mask = 0777
```

Oder ich möchte ein Verzeichnis, was nur von einer Mitarbeitergruppe ansprechbar ist:

```
[Objekt]
        comment = Objektteam Heynitz
        path = /usr/local/samba/Objektteam
        read only = no
        create mask = 0660
        directory mask = 0770
        force group = objekt
        valid users = @objekt
```

4.1.6 Drucker

Da bei uns das 'normale' lpr-System läuft, gestaltet sich die Freigabe der Drucker recht einfach:

```
[printers]
   comment = All Printers
   path = /var/spool/lpd
   browseable = yes
   public = yes
   guest ok = yes
   printable = yes
```

Damit werden alle dem Server bekannten Drucker (/etc/printcap) auch unter SAMBA sichtbar.

4.2 Und die Clients

Win2000 kann echt zickig sein und erfreut oft mit sinnfreien Fehlermeldungen. Daher sollten einige Dinge beachtet werden:

1. auf dem Server zuerst einen Maschinenaccount anlegen

   ```
   root# /usr/sbin/useradd -g 100 -d /dev/null -c "machine nickname" -s /bin/false machine_name$
   root# passwd -l machine_name$
   root# smbpasswd -a -m machine_name
   ```

2. root muß als SMB-User angelegt sein (smbpasswd -a root), hierbei kann ein anderes Passwort als das UNIX-Passwort verwendet werden.

3. es dürfen keinerlei Laufwerke gemappt sein

4.3 SWAT

SWAT *Samba Web Administration Tool* richtet sich vor allem an Anwender, die (noch) ihre Probleme mit der Kommandozeile haben.

SWAT ermöglicht, via Web-Oberfläche die vielfältigen Optionen der SAMBA-Konfiguration auf einfache Art und Weise zu beeinflussen.

4.4 WebMin

Eine weitere Möglichkeit der grafischen Administration von SAMBA stellt WebMin bereit.

Da ich weder SWAT noch WebMin nutze, gehe ich auf diese Programme nicht weiter ein.

4.5 Fehlersuche

Normalerweise läuft SAMBA auf Anhieb. Falls es dennoch Probleme geben sollte, sollte man in dieser Reihenfolge auf Fehlersuche gehen:

1. Die einfachsten Sachen sind manchmal das größte Problem. Findet der *smbd* seine Konfiguration? Mittels `testparm` ist das einfach zu testen, es sucht in dem einkompiliertem Standardverzeichnis nach der `smb.conf` und prüft ihren Inhalt.

2. Steht das Netz? Testen Sie das mit `ping` unter Verwendung der IP-Adresse.
3. Klappt die Namensauflösung? Testen Sie mit `ping`, nun aber mit den Rechnernamen.
4. Sie können auf dem Server die Freigabeliste testen: `smbclient -L server`
5. `nmblookup -B server __SAMBA__` testet die NetBios-Namensauflösung und sollte die IP des Servers liefern
6. Suchen Sie vom Server aus die Clients:

   ```
   nmblookup -B IP-des-Client '*'
   nmblookup -B IP-des-Client Client
   ```

   Beide Abfragen sollten mindestens die IP des Clients liefern, wenn nicht, sind die Clients nicht richtig konfiguriert.

5 Tuning

SAMBA kann vom SOHO-System bis zu richtig großen 'Trümmern' laufen - ein Patentrezept fürs Tuning fällt da schwer. Außerdem ist es oft auch eine Entscheidung mit Nebeneffekten. Hier ein paar Tricks:

5.1 diverse Tipps

5.1.1 Access

Für Shares, auf denen Access-Dateien liegen, kann dieser Schalter hilfreich sein:

```
kernel oplocks = no
```

Allerdings bedeuted dies oft auch einen Geschwindigkeitsverlust.

5.1.2 Netzwerk

Die Broadcasts können in größeren Netzen recht schnell zur Plage werden. Abhilfe schafft hier WINS, so wie auch in der Konfiguration verwendet.

5.1.3 Verbindungen

Nicht aktive Verbindungen können getrennt werden, das spart Ressourcen auf dem Server. Dazu dient `deadtime`

5.1.4 Cache

Mit `getwd cache = yes` kann ein Cache für des Inhaltsverzeichnis eingeschaltet werden.

5.1.5 WinXP

WinXP testet zuerst die Erreichbarkeit des Servers auf Port 80. (WebDAV). Falls kein Apache installiert ist, liefert der TCP/IP-Stack ein 'port unreachable', falls er läuft, eine Antwort. Falls man aber z.B. seinen Server in eine DMZ stellt und Anfragen DROPt (was selten gut ist), wartet XP bis zum Timeout.

Abhilfe:

1. REJECT statt DROP
2. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxDAV auf 0 setzen

5.1.6 Fehlermeldungen W2K bei Beitritt in Domäne

```
"Bei dem Versuch der Domäne "" beizutreten, trat folgender Fehler auf.
Die angegebende Anmeldeinformation stehen mit vorhandenen
Anmeldeinformationen im Konflikt"
```

Bestehende Verbindungen (gemappte Laufwerke) trennen.

```
"Das verwendete Konto ist ein Arbeitsstationskonto. Verwenden
Sie Ihr normales Benutzerkonto oder lokales Benutzerkonto, um
auf diesen Server zuzugreifen"
```

Domänen-Anmeldung geht nur als Administrator.

Ansonsten sind viele Tipps und Tricks zum Tuning in der Doku zu SAMBA, insbesondere in Speed.txt, genannt.

Was die Geschwindigkeit allgemein anbelangt:

```
,----[  http://www.heise.de/newsticker/data/ps-15.10.03-000/  ]
| Samba 3 schneller als Windows 2003 Server
|
| Die britische IT Week hat Ergebnisse von Benchmarks veröffentlicht, die sie mit
| Samba 3 und Windows 2003 Server unternommen hat. Demnach ist das neue Samba 3
| bis zu zweieinhalb mal schneller als Windows 2003 Server. Damit hat sich die
| freie Server-Software laut IT Week sogar noch gesteigert: Im vergangenen Jahr
| war in ihrer Untersuchung die Version 2 im Vergleich zum Windows 2000 Server
| lediglich doppelt so schnell.
|
| Über die eigentliche Untersuchung erfährt man bei IT Week nicht allzu viel. Als
| Testsoftware kam der Ziff-Davis NetBench zum Einsatz. Als Server diente ein mit
| 900 MHz getakteter Pentium III mit IDE-Platte und 512 MByte RAM. Weitere
| Untersuchungen zu diesem Thema dürften nicht lang auf sich warten lassen.
| Bisher gelang es beiden Seiten, sehr unterschiedliche Aussagen über die
| Performance zu treffen, so zuletzt Veritest im Auftrag von Microsoft. (ps/c't)
`----
```

6 Linux als Client

Natürlich ist auch der umgekehrte Weg möglich, also Linux nutzt Freigaben von Windows-Rechnern.

Ich nutze es allerdings eher selten, für mich ist der Linux-Server zentraler Punkt des Netzes, nicht zuletzt wegen der Sicherungsmöglichkeiten.

Wer also Freigaben von Windows nutzen will, kann mittels `smbclient` eine FTP-ähnliche Verbindung zu einer Windows-Freigabe machen bzw. mittels `smbmount` Freigaben direkt mounten. Man kann mittels `smbclient` auch Drucker nutzen und auch Backups von Shares erstellen.

Falls die Clients Windows NT oder höher haben, kann man dort auch sehr einfach einen *lpr*-Dienst installieren und angeschlossene Drucker einfacher und unkomplizierter als via `smbclient`nutzen, sie verhalten sich dann wie ein Printserver. Allerdings meint manchmal der Druckertreiber von Windows, die Druckdaten verschönern zu müssen, was eher selten gut ausgeht. Daher installiere ich solche Drucker auf Windows-Seite als 'Nur-Text-Drucker'.

7 Sicherheit - eine schlechte und eine gute Nachricht

Die schlechte Nachricht: Auch SAMBA ist hin und wieder gut für Überraschungen.

Daher ist es notwendig, sich über die Sicherheit seiner Konfiguration Gedanken zu machen. In so mancher kleinen Firma werkelt ein SAMBA gleichzeitig mit anderen Diensten auf einem Gateway zum Internet, und nicht nur in kleinen, wie ich aus eigener Erfahrung weiß.

Daß das nicht optimal ist, liegt auf der Hand. Andererseits will man nicht immer Geld und Arbeit in 2 Systeme stecken. Kurz und gut: kann ja mal passieren.

Die gute Nachricht: auch sowas kann man weitestgehend sicher bekommen.

Erste Maßnahme sollte sein:

```
interfaces = eth0 192.168.1.1/24 lo
bind interfaces only = yes
```

Der *nmbd* lauscht allerdings weiter auf Broadcasts, verwirft jedoch Pakete mit fremder SRC. Um ganz sicher zu gehen, kann man mit iptables nachhelfen:

```
iptables -A INPUT -i ippp0 -p TCP --dport 137:139 -j --reject-with tcp-reset
iptables -A INPUT -i ippp0 -p UDP --dport 137:139 -j --reject-with icmp-port-unreachable
```

7.1 Und wer SMB via Internet anbieten will?

Der sollte sich das nocheinmal überlegen. Eine einfache Alternative wäre SSH/SCP, eine umfangreichere Variante wäre ein VPN.

8 Ausblick

Ich habe hier eine konkrete Installation eines SAMBA 2.2.x beschrieben, allerdings ohne viel technische Raffinessen.

- Die Userverwaltung in größeren Netzen kann man zentralisieren, das Mittel der Wahl ist LDAP. Auch SAMBA kann das nutzen.

- Drucker lassen sich bequemer über CUPS verwalten, man kann auch Druckertreiber für die Clients auf dem Server bereitstellen, die sich die Clients bei Bedarf dort holen.
- Seit ein paar Wochen ist SAMBA 3 verfügbar. Es bringt u.a. ADS-Support, Unicode, besseren Druckersupport und anderes.

9 weiterführende Quellen

`http://samba.org/`

`http://Samba.SerNet.DE/info.html/`

`http://de.samba.org/samba/ftp/docs/htmldocs/samba-pdc-faq.html/`

`http://us2.samba.org/samba/ftp/docs/htmldocs/Samba-PDC-HOWTO.html/`

`http://samba.idealx.org/`

Newsgroup de.comp.os.unix.networking.samba