http://www.therealgang.de/

Titel :	ISFS
Author :	David Biermann
Kategorie :	WEBSERVER-ALLGEMEIN

Akademie der Saarwirtschaft

ISFS

Dozent:
Walter Neu

Script created by:
David Biermann

SAMBA

Kapitel 1

Samba lässt jeden Unixrechner in der Netzwerkumgebung von Windows erscheinen. Der Client merkt keinen Unterschied on Windowsserver oder Unixserver.

Eigenschaften von Samba:
- Dateifreigaben können einfach erstellt werden
- Benutzer kann Dateien im eigenen HOME und in freigegeben Verzeichnisse ablegen
- Drucker, die unter Unix ansprechbar sind, können als Netzwerkdrucker in Windows angesprochen werden.

Fähigkeiten von Samba:
- WINS Server kann einfach eingerichtet werden
- Computersuchdienst (stabiler Server)
- Diagnosewerkzeug (effektive Werkzeuge zur Vereinfachung von Fehlersuche)

Vorteile von Samba, die von Unix geerbt wurden:
- entfernte Administration (von Kommando - Zeile aus)
- zentrale Konfiguration(eine einzige Textdatei)
- Stabilität von Unix

SMB – Server Message Protokoll
Client/Server Protokoll, das nach dem Frage/Antwort Prinzip arbeitet. Client stößt Aktion durch Anfrage an. Server gibt Antwort/Fehlermeldung zurück.

Konfiguration
Samba wird mit der Datei smb.conf konfiguriert.

UNC Pfad
\\rechnername\pfad\dateiname (Universal naming convention)

Kapitel 2

Bestandteile von Samba:

Servertools
Die beiden Dämonen smbd und nmbd stellen die eigentliche Dienste zur Verfügung und sind zum Starten von Samba notwendig.

SMBD = zentraler Serverprozess, der für die Datei+ Druckerfreigaben zuständig ist. Es gibt mehrere smbds im System. Einer hört auf den Port 139 und nimmt neue Verbindungen entgegen. Jede Verbindung stößt einen neuen smbd Prozess an. Sollen Verbindungen getrennt werden, müssen nur mit `smbstatus` die Prozessnummer der zuständigen smbds erfragt und gelöscht werden.

NMBD = ist für die NetBios Namens, - und Datagrammdienste zuständig. Dieser Prozess reserviert beim Start von Samba die entsprechenden NetBios Namen. Er kann WINS Server sein und ist für den Computersuchdienst zuständig.

SMBPASSWD = serverseitige Pflege der verschlüsselten Passwörter.

Clientseitige Tools

Samba kann nicht nur als Server, sondern kann auch als Client auf Windows Rechner zugreifen.

- *smbclient:* Programm, mit dem man auf Freigaben von NT-Rechner zugreifen kann z.B. drucken auf von NT zur Verfügung gestellten Drucker. Mit smbclient kann auch die Liste der Server im Netz erfragt werden.
- *nmblookup*: Diagnosetool für Namensauflösung. (Wenn 2 PCs sich nicht finden können, kann mit diesem Befehl deren Versuche sich zu finden, nachgestellt werden.)
  WINS Server können befragt werden und ein **NetBios Status Request** abgefragt werden.
  (analog zu nbtstat unter Windows)

Kapitel 3

NetBios

Net Bios = Softwareschnittstelle zur Kommunikation von Rechner.
Wenn Windows Rechner Netzwerklaufwerke verbinden, sich gegenseitig in der Netzwerkumgebung sehen oder Drucker freigeben, funktioniert ihre Kommunikation untereinander über NetBios. Mit dieser Schnittstelle werden Programme unterschiedlicher Dienste zur Kommunikation zur Verfügung gestellt.

NBT Standard beschreibt derzeit drei Netzwerkdienste:
Namensdienst, Datagrammdienst und Sitzungsdienst.

NetBios Dienste

Dienst	Protokoll	Port	Sambaprozess
Namensdienst	UDP	137	Nmbd
Datagrammdienst	UDP	138	Nmbd
Sitzungsdienst	TCP	139	smbd

- *Namensdienst*: gegenseitige Identifizierung der Rechner im Netz. Wollen Anwendungen zwischen Rechner im Netz kommunizieren, müssen sie sich zuerst gegenseitig identifizieren können. Dazu werden 16 Byte lange Namen gebraucht. Von den 16 sind 15 nutzbar, das letzte Byte kennzeichnet den Ressourcentyp des Namens. NetBios Namen existieren im einen, flachen Namensraum, d.h es gibt kein Äquivalent zu Domänen Bezeichnungen. Namen dürfen alphanumerisch sein und gewisse Sonderzeichen enthalten. Jede Anweisung kann für sich beliebig viele Namen reservieren. Unter einem Namen werden Verbindungen aufgebaut und Daten ausgetauscht. Die Reservierung von Namen gilt für Clients als auch für Server.
Wollen 2 Anwendungen per NetBios kommunizieren, muss der Server zuerst seine Bereitschaft, Verbindungen entgegenzunehmen, kundtun. Dazu reserviert er im Netz per Broadcast seinen Namen, so dass alle im Subnetz mithören. Dieser Vorgang, (Reservierung per Broadcast) passiert insgesamt 3x. Erfolgt daraufhin kein Protest, so sieht der Server seinen Namen als reserviert an. Verfahren der Namensreservierung bei Clients identisch, allerdings muss der Client zuerst die MAC - Adresse des Servers herausfinden.

- *Datagrammdienst*: Dienst, um schnelle, einfache Nachrichten zu versenden. Es erfolgt keine Verbindung zwischen den Computern beim Versand der Daten (~ ist ein verbindungslos – orientierter Dienst). Daten werden in einzelne Pakte zerlegt und an den Zielrechner ausgeliefert. Pakete können verloren gehen, in vertauschter Reihenfolge oder mehrfach ankommen. U.U erhält der Sender auch nicht einmal eine Benachrichtigung über verlorengegangene Pakete. Einziger

Vorteil: geringer Aufwand beim Verschicken von Daten und gleichzeitiges Verschicken von Datagrammen an mehrere Rechner.

- *Sitzungsdienst*: entspricht Telefonverbindung, zwischen den beiden kommunizierenden Applikationen besteht eine Verbindung ⇔ Datagrammdienst.
Es wird eine NetBios Sitzung vereinbart. Daten kommen mit dem ~ auf jeden Fall richtig und auch in der richtigen Reihenfolge an. Falls nicht, erhält die versendende Applikation eine entsprechende Fehlermeldung. Dieser Zuverlässigkeit steht ein höherer Aufwand beim Sitzungsaufbau und Abbau gegenüber.

<u>NetBios Implementationen</u>
NetBios kann mit unterschiedlichen Protokollen implementiert werden: NetBeui, IPX oder TCP/IP.

- *NetBeui*: Client findet Server nur über Broadcasts. Der Server, der sich für den gesuchten Namen verantwortlich fühlt, antwortet, nachdem er seine MAC - Adresse ausgelesen hat. Mit NetBeui können nur Rechner miteinander kommunizieren, die in der gleichen Broadcastdomäne liegen.

- *TCP/IP*: Client muss IP des Servers herausfinden. Dies kann über Broadcast im lokalen Netz geschehen. Befindet sich der Rechner im gleichen Subnetz, kann direkt eine ARP Anfrage nach der MAC - Adresse ausgelöst werden. Andernfalls muss der entsprechende Router anhand der Routingtabelle herausgefunden werden und dann dessen Mac-Adresse per ARP festgestellt werden.

ARP steht für Adress Resolution Protocol und läuft in 4 Schritten ab:
1.Datenpaket an Ethernet Netzwerk Schnittstelle der eigenen Station übergeben. Sie sucht MAC - Adresse in der eigenen Tabelle. Ist ein gültiger Eintrag vorhanden, wird ein Paket mit der gefundenen Adresse versehen und gesendet.
2.Ist in der eigenen Tabelle kein gültiger Eintrag vorhanden, wird ein ARP – Broadcast mit der IP – Adresse des Zielhosts erzeugt und gesendet.
3.Alle im LAN erhalten das Broadcast Paket und vergleichen die darin enthaltene Ziel IP mit ihrer eigenen. Die Station, mit der gesuchten IP - Adresse sendet ein ARP – Paket, das die gesuchte Ethernet-Adresse enthält an den anfragenden Rechner.
4.Anfragender Host trägt nach dem Empfang MAC – Adresse in seine Tabelle ein und sendet das IP Paket direkt an den gesuchten Host.

Kapitel 5

NetBios Namensanfrage wird mit nmblookup ausgelöst. Paket wird an die Broadcastadresse im lokalen Subnetz gesendet. Nmblookup entnimmt konkrete Broadcastadresse der Zeile `interfaces = smb.conf`. Unter Windows ist eine isolierte Namensanfrage nicht möglich. Es muss eine Verbindung aufgebaut werden. Eine Anzeige der reservierten Namen erfolgt mit der Operation **Node Status Request**. Ein Rechner hat gleich mehrere Namen für sich reserviert. Jeder Name steht für eine andere Anwendung. (Unterscheidung am 16.Byte)
Es gibt NetBios Gruppen, - und Einzelnamen. Einerseits müssen bestimmte Dienste benannt werden, andererseits müssen manche Anwendungen mit mehr als einem Partner gleichzeitig kommunizieren.

<u>*Einzelname*</u>: existieren nur ein einziges Mal im gesamten Netz.
- `computername<00>` = Client tut seine Existens kund, dient zur eindeutigen Identifizierung
- `computername<20>` = Name für Serverdienst, Funktion eines Servers

- benutzer<03> = Anmeldung des Nachrichtendienstes des Rechners
- arbeitsgruppe<1d> = LMB

Gruppenamen: existieren mehrfach im Netz, „Broadcast für Arbeitsgruppe“. Existieren Gruppennamen, können unter diesem Namen alle Rechner dieser Arbeitsgruppe mit einem Datagramm erreicht werden.

- arbeitsgruppe>00> = Zugehörigkeit zu einer Arbeitsgruppe
- arbeitsgruppe<1c> = der Domän Logon Server reserviert diesen Namen für sich
- arbeitsgruppe<1e> = alle Rechner, die LMB werden können, reservieren diesen Namen für sich
- .._MSBROWSE_.<01> = alle LMBs um sich gegenseitig zu finden

Kapitel 6

NetBios: mit Hilfe von ~ sind Rechner im Netz ansprechbar und können verschiedene Dienste anbieten.
Arbeitsgruppe: Liste von Rechner, nur als reines Transportmedium mit NetBios zu tun.
Domäne: gemeinsam genutzte Benutzerdatenbank von Rechner

Local Master Browser
Rechner, der die Netzwerkumgebung pflegt, wird gewählt, nicht bestimmt, Wahl wird von dem Rechner angestoßen, der als 1. merkt, dass es kein solcher LMB gibt.
Will ein Rechner die Netzwerkumgebung anschauen, kontaktiert dieser den LMB über den NetBios Namen arbeitsgruppe<1d>. Server, die angezeigt werden wollen finden den LMB auf die gleiche Weise.
Wahl zum LMB: per Datagramm an Gruppennamen <arbeitsgruppee<1e>
Kriterien:
- OS Level
- Betriebssystem, dass besser ist wie ein anderes verschickt Wahlpaket mit Parameter
- Uptime, Rechner, der am längsten läuft „gewinnt“
- NetBios Name, alphabetischer Reihenfolge

Kapitel 7

NetBios über Subnetzgrenzen
Rechner die hinter Routern liegen, können über Broadcast nicht erreicht werden, denn Broadcasts verbleiben nur im Subnetz.

Möglichkeiten der Namensauflösung: Broadcast, LMHOSTS, WINS

1.LMHOSTS: einfachste Weg Namensauflösung über Subnetzgrenzen hinweg zu realisieren. Sie ist eine statisch zu verwaltende Datei und liegt unter /etc/hosts. Der Zusatz von #PRE bewirkt direktes Verwenden des Werts in der LMHOSTS. Ohne den Zusatz wird zuerst eine konventionelle Namensauflösung durchgeführt. Der Nachteil der ~ ist, dass sie auf jedem Rechner statisch zu pflegen ist.

2. WINS Server: dynamische Datenbank auf zentralem Server zur Pflege der NetBios Namen. Jede NetBios Applikation muss sich im Netz mit eigenem Namen anmelden.

IP dieses Servers muss jedem Rechner mitgeteilt werden. Dies geschieht bei Samba durch den Eintrag `wins server = <ipadresse>` im Abschnitt `<global>` in der `smb.conf` .Sobald ein Rechner die IP - Adresse des WINS Servers kennt, ist es egal ob sich dieser im gleichem Subnetz befindet oder nicht. Namenreservierung passiert nicht mehr über Broadcast sondern per gerichteten UDP – Paket an den WINS Server. Router leitet gerichtetes Paket wie jedes andere Paket an den WINS Server weiter. Möchte ein Rechner einen Namen reservieren, der schon vergeben ist, fragt der WINS Server nach, ob der Name noch gebraucht wird. Wird der Name noch gebraucht, bekommt der anfragende Rechner eine Ablehnung. Wird der Name nicht mehr gebraucht, oder bekommt der WINS Server keine Antwort so bekommt der anfragende Rechner eine positive Benachrichtigung. Diese Vorgehensweise ist dafür gedacht, um mit abgestürzten Rechner sauber umgehen zu können. Die Anfrage an den WINS Server erfolgt mit nmblookup.
Z.B `nmblookup 192.168.1.5 Samba = WINS Server`
Server, der die IP 192.168.1.5 hat, wird nach dem Namen Samba befragt.

Kapitel 8

NetBios Anwendungen = Windows Programme um Laufwerke mit Server zu verbinden. Die gesamte Netzwerkverbindung gehört ebenfalls zu den NetBios Anwendungen.
System schaut im NetBios Namenscache nach, ist ein WINS Server konfiguriert, wird dieser befragt. Kann der Name nicht aufgelöst werden, so wird eine Broadcast Anfrage ausgelöst. Es wird in der LMHOSTS Datei nachgesehen. Falls DNS Auflösung für NetBios in den TCP/IP Eigenschaften aktiviert ist, wird das Auflösungssystem für TCP/IP Anwendungen übergeben. Namenseinträge in LMHOSTS werden erst nach den WINS und Broadcast Timeouts berücksichtigt.

TCP/IP Anwendungen = Anwendungen, die es nur in der TCP/IP Protokollfamilie gibt. Namensauflösung funktioniert etwas anders als bei NetBios Anwendungen. Es wird in der LMHOSTS nachgesehen, ist ein DNS Server konfiguriert, wird dieser befragt. DNS Name wird an die NetBios Namensauflösung übergeben.

Samba kann sowohl als Client, als Server und auch als Domänenmitglied auftreten. Als Client und als Server muss Samba Namen auflösen. Samba kennt wie Windows 4 Mechanismen um dies zu tun: Broadcast, WINS, LMHOSTS und die Unix Namensauflösung.

Kapitel 9

Wenn eine einheitliche Arbeitsgruppe über Subnetzgrenzen hinweg gewünscht wird, muss ein weiterer Dienst installiert werden: DMB
Domän Master Browser = Rechner, der die Serverlisten von allen LMBs einsammelt und auf Anfrage wieder herausgibt. Der DMB wartet nur passiv darauf, dass ein LMB sich mit ihm synchronisieren will. Die LMBs haben die Aufgabe sich regelmäßig danach zu erkundigen, wo der DMB sitzt und mit ihm diesem die Serverlisten abzugleichen. Damit ein Samba Server die Aufgaben eines DMB übernehmen kann, ist innerhalb der smb.conf der Parameter `domain master = yes` in der [global] Section zu setzten.

Kapitel 10

Virtuelle Samba Server
Einen einzigen Server auf einer Maschine laufen zu lassen, nutzt einen PC heute bei weitem nicht mehr aus. Ein Samba Server ist in der Lage mehrere Identitäten gleichzeitig anzunehmen. Zur

Serverkonsolidierung kann es nötig sein, unter mehreren Namen in der Netzwerkumgebung zu erscheinen. Eine andere Konfiguration ist die Einbindung von virtuellen Samba Servern in eine Hochverfügbarkeitsumgebung.

Kapitel 12

SMB Sitzungen

Um Fehlerdiagnose zu betreiben, ist das Wissen um die genaue Fehlerursache wertvoll.

NetBios Namensauflösung

Namen des Servers eingeben beim Aufbau einer Verbindung:
- Doppelklick in der Netzwerkumgebung auf einen Rechner
- von der Kommandozeile aus `net use h: \\server\freigabe`
- Netzlaufwerke verbinden
- ausführen im Startmenü \\server = Anzeige der Freigaben des Servers

TCP Verbindungen

Wenn die Adresse, zu der verbunden werden soll klar ist, wird eine TCP Verbindung zu Port 139 des Servers aufgebaut. Um vorhandene Verbindungen sich anzeigen zu lassen, gibt es das Werkzeug `netstat`. Ob die TCP Verbindung geklappt hat, prüft man mit `telnet <ip> 139`.
=> Entweder Fehlermeldung oder Verbindungen sind ESTABLISHED.

NetBios Sitzungen

Alle Anwendungen haben für sich Namen reserviert und sind unter der IP – Adresse des Rechners und dem TCP Protokoll auf dem Port 139 zu erreichen. Anhand des TCP _ Verbindungsaufbaus ist nicht klar, welche Serverapplikation angesprochen werden soll. Die Unterscheidung wird durch den Servernamen getroffen, der in der TCP - Verbindung als erstes übertragen wird.

Negotiate Protocol

Innerhalb einer NetBios Sitzung wird eine SMB Sitzung schrittweise aufgebaut:
Die erste Anfrage die der Client an den Server schickt, ist ein **Negotiate Protocol Request**. Er schickt eine Liste der Protokollvarianten, die er beherrscht. Der Server wählt eine Protokoll für die weitere Kommunikation aus der Liste aus und schickt Index des jeweiligen an den Client zurück. Außerdem werden zwei weitere Einstellungen verschickt:
- die Zugriffssteuerung auf Benutzer - oder auf Freigabeebene (`security = share` bedeutet auf Freigabeebene und `security = user` auf Benutzerebene)
- der Zeitpunkt, zu dem der Benutzer ein Passwort liefern muss (direkt beim Session Setup oder erst danach beim Tree Connect)
- Verwendung von Klartextpasswörtern oder verschlüsselten Passwörter (werden verschlüsselte Passwörter verwendet wird eine Herausforderung für das Challenge Response mitgeschickt)

Es ist nicht möglich für einige Benutzer Klartextpasswörter und für andere verschlüsselte Passwörter zu verwenden.

Session Setup

Nachdem Abhandeln der Protokollversion wird vom Client ein Session Setup verschickt. Darin enthalten ist der Benutzername des Clients und falls vom Server vorher verlangt(security = user) auch das Passwort. Damit ist der Server in der Lage die Identität des Benutzers festzustellen.

Tree Connect
Als letztes legt der Client fest, welche Freigabe er ansprechen will. Der Entsprechende Aufruf heißt ~. Wenn security = share angegeben wurde, wird der Server an dieser Stelle das Passwort überprüfen.

Kapitel 13

Rechte an Freigaben
Ist bei Samba security = users gesetzt, so hat der Server die Möglichkeit anhand des angemeldeten Benutzers Zugriffsrechte zu vergeben und zu verweigern. Wenn bei der Einstellung einer Freigabe keine Parameter für die Zugriffsrechte gesetzt sind, hat jeder korrekt angemeldete Benutzer Leserecht. Mit den Optionen zur Rechtevergabe an Freigaben hat man die Möglichkeit einzelnen Benutzer und ganzen Unixgruppen Rechte zu geben oder zu nehmen.

Alle Benutzer haben gleichen Zugriff
[projekt] path = /data/projekt = alle angemeldeten Benutzer mit Name und Passwort haben Lesezugriff auf die Freigabe. Schreibrecht vergibt man mit `writeable = yes`.

Einige Benutzer haben gleichen Zugriff
`[projekt] path = /data/projekt valid users = mueller , uhl` (Einschränkung auf Benutzer Müller und Uhl) Optional kann ihnen auch noch Schreibrecht gegeben werden, writeable = yes
Root muss wie jeder Benutzer in die Liste aufgenommen werden!
Mit valid users können auch ganze Unixgruppen in den Zugriff aufgenommen werden (@ Zeichen)
z.B. `valid users = root, @users`

Einige Benutzer haben Leserecht andere Schreibrecht
```
[projekt] path = /data/projekt
valid users = @users, @admins
write list = @admins
```

Kapitel 14

Zugriffsrechte im Dateisystem
Benutzer muss sowohl durch eine Freigabe -, als auch durch Dateirechte zu Operationen berechtigt sein. Nach erfolgreichem Verbund mit der Freigabe nimmt der Benutzer seine ganz normalen Rechte als Unix User wahr. Will ein Benutzer in eine Datei schreiben, muss ihm dies sowohl durch die Freigabe als auch durch die Dateisystemrechte erlaubt sein. Von Samba vergebenen Rechte können darunter liegende Unixrechte nicht erweitern. Die Einschränkung durch Unixrechte ist ein wichtiges Prinzip von Samba. Im Dateisystem implementiert Samba keine eigenen Zugriffskontrollen, sondern verlässt sich auf die Unixmechanismen.
2 Gründe
- Zugriffsrechte sind im Betriebsystem bereits vorhanden implementiert
- es ist nicht möglich Samba ACLs synchron mit dem Unix – Dateisystem zu halten (falls sich Verzeichnisstrukturen ändern, wie soll dann die Samba ACLs angepasst werden können?!)

DOS Attribute
Diese Attribute sind Eigenschaften von Dateien, die es in dieser Form unter Unix nicht gibt. Insgesamt kennt DOS 4 verschiedene Attribute, die für Dateien vergeben werden können:
- `Read – only` (Inhalt kann nur gelesen werden, nicht geschrieben und gelöscht werden)
- `System` (für Betriebssystemzwecke)

- `Hidden` (diese Dateien werden mit dem Kommando dir nicht angezeigt)
- `Archiv` (wird bei jedem Schreibzugriff gesetzt)

Zugriffsrechte im Dateisystem

user	group	others
r w x	r w x	r w x

// Windows

schreibgeschützt
archiv
system
versteckt

SCHREIBGESCHÜTZT
user
r - x --> schreibgeschützt
r w x --> darf schreiben

// Reihenfolge ASW (Archiv,System,Versteckt)--> Execute Bit bei (User,Group,Others)

Kapitel 16

Opportunistic Locks (Oplocks)
Dateizugriffe über Netzwerk deutlich langsamer als auf einer lokalen Festplatte.
Zugriffe von Clients auf Freigaben müssen koordiniert werden, da ein Cache auf Netzwerkdateien nicht davon ausgehen kann, die Datei alleine zu benutzen.
Opportunistic Locks (Oplocks) sind Mechanismen, mit dem Clients erlaubt werden kann, Dateiinhalte zu cachen. Mit einem Oplock bekommt der Client eine Datei solange exklusiv für sich, bis der Server ihn auffordert, die Änderungen zurückzuschreiben und die Sperre freizugeben. Oplock ist die Zusage, dass niemand sonst auf die Datei zugreifen kann. Damit muss ein Client weder bei jedem Lesezugriff den Server befragen, noch muss er jeden Schreibzugriff unverzüglich an den Server liefern. Wenn ein weitere Client auf die Freigabe zugreifen möchte, schickt der Server dem Client A ein so genanntes *Oplock Break*. Dies ist die Anweisung sämtliche Änderungen zurückzuschreiben und den Schreibcache auf diese Datei in Zukunft auszuschalten. Erst nachdem die Änderungen zurückgeschrieben wurden, kann Client B auf die Datei zugreifen. Die Änderungen sind sofort sichtbar. Dieses Schema funktioniert innerhalb von Samba hervorragend. Sobald Unix Prozesse ebenfalls auf Dateien zugreifen müssen, die von Samba freigegeben wurden, gibt es Probleme.

Kapitel 17

Benutzerauthentifizierung muss vor allem 2 Dinge leisten:
- Benutzer muss beweisen, dass er sein Passwort kennt
- Authentifizierungsprotokoll kann es dabei ermöglichen, dass das Passwort nicht übertragen werden muss.

Es gibt 2 Verfahren zur Verschlüsselung:
- **symmetrische Verschlüsselung**
  Nachricht wird zerstückelt, so dass niemand sie mehr lesen kann (außer jemand kennt das Verfahren, mit dem verschlüsselt wurde), es existiert zu der Verschlüsselung ein Gegenstück, das wieder die originale Nachricht herstellt. D.h es muss nicht unbedingt das Verfahren bekannt

sein, mit dem verschlüsselt wurde, sondern nur der Schlüssel.

- asymmetrische Verschlüsselung
Schlüsselpaar: private + public key
Mit Hilfe von public key Nachricht verschlüsselt
Empfänger kann mit private key Nachricht wieder entschlüsseln
Schlüsselpaar wird mit Hilfe von Programmen erstellt.

Challenge Response Verfahren
Bevor Verbindung zum Server aufgebaut wird, muss Client sich am Server anmelden mit Benutzername und Passwort. In der Antwort auf das Negotiate Protocol Response schickt der Server dem Client eine Zufallszahl, *die Herausforderung* genannt wird. Mit ihr wird nun nicht das Passwort verschlüsselt, sondern umgekehrt. Das Passwort wird als Schlüssel benutz um die Herausforderung zu verschlüsseln. Die mit dem Passwort verschlüsselte Herausforderung schickt der Client in der Session Setup zusammen mit Benutzernamen an den Server. Der Server liest nun aus seiner Benutzerdatenbank das Passwort des Benutzers aus und entschlüsselt damit die Herausforderung wieder und vergleicht das Ergebnis mit der Zufallszahl, die er dem Client geschickt hat und sich gemerkt hat. Somit kann der Server simpel überprüfen, ob das Passwort des Benutzers stimmt, oder nicht.

Kapitel 19

19.2 Arten der Anmeldung
19.3
UID keine Gültigkeit über Rechnergrenzen (Linux)
SID hat Gültigkeit über Rechnergrenzen (Windows)
RID vergleichbar einer Linux UID
SID - RID --> über Rechnergrenzen nutzbar

Kapitel 2

Ablauf einer http (hypertext transfer protocol) Verbindung
Kommunikation zwischen Client und Webserver erfolgt durch den Austausch von http Nachrichten. Client stellt über TCP Verbindung auf Port 80 zum Webserver her.
Danach schickt der Client eine Anfrage(Request) an den Server, in dem er die Informationen spezifiziert, die er einsehen möchte. Der Request enthält u.a die Art der Anfrage(Methode),den URL(Universal Resource Locater) und die Protokollversion (http/1.0 und http/1.1).
Auf dem Server wird die Anfrage registriert. Dieser durchsucht daraufhin sein Dateisystem nach der geforderten Datei und lädt sie. Server schickt Antwort (Response) mit gewünschten Informationen und MIME Typ oder Fehlermeldung. Aus dem MIME Type schließt der Client, was er mit den Informationen anfangen soll, z.B im Bowser anzeigen. Die Verbindung wird direkt nachdem Senden des Response beendet (Leitungskapazität wird geschont).
Deshalb bezeichnet man http auch als ein zustandsloses Protokoll.
(Gegenteil ist ein zustandsorientiertes Protokoll z.B ftp.)

HTTP Requests
Sie werden durch die Angaben von Methode, URL und den Requests Header Feldern bestimmt. Zu jeder Methode gehört ein Ziel URL. Dabei muss der Client eine absolute URL angeben. Request Header haben folgende Struktur:

Methode URL http/Version

z.B GET http://www.web.de/verzeichnisse/index.html HTTP/1.1

HTTP Methoden
GET Methode ist die wichtigste Methode. Sie dient zur Anforderung eines Dokuments oder einer anderen Quelle. Quelle wird dabei durch den URL identifiziert. Man unterscheidet conditional GET und partial GET.

POST Methode nimmt den umgekehrten Weg wie die GET Methode. Sie übermittelt in erster Linie Formulareingaben und auch Kommentierung bestehender Quellen.

PUT Methode erlaubt die Modifizierung bestehender Quellen bzw. Erzeugung neuer Daten auf dem Server.

HTTP Response
Struktur aus Header und Nachrichten – Body ist bei Request und Response gleich, trotz unterschiedlichen Informationen:
http/Version ; Status Code und Reponse Zeile
http/1.1 200 OK
Der Server übermittelt zunächst die http Version der Nachricht, dann eine Statusmeldung. 200 OK bedeutet das keine Fehler aufgetreten sind. Wichtig für die weitere Bearbeitung für den Client sind die beiden Einträge Content Type (beschreibt den MIME Type der im Datenbereich übermittelt wird) und Content Length (Länge der Daten in Byte)

Response Codes
Die Antwort des Servers beinhaltet die Stauszeile und Response Header. Die Statuszeile wiederum führt die Protokollversion, den Status Code und den Reasons Phrase auf. Beim Status Code handelt es sich um einen dreistelligen Integer Wert, der dem Client wichtige Informationen über Verfügbarkeit, erfolgreiche Bearbeitung oder auch Fehlermeldungen liefert.
Die Reason Phrase enthält die Klartext Bezeichnung der Meldung. Bekannte Fehlermeldung ist die 404 „File not found“

1xx Informelle Meldungen, 2xx Erfolgsmeldungen, 3xx Weiterleiten, 4xx Clientfehler, 5xx Serverfehler

Kapitel 6

Apache ist nicht monolithisch sondern modular aufgebaut. Der Kernserver kennt nur die grundlegenden Eigenschaften eines Webservers. Zusätzliche Features werden mit Hilfe von Modulen eingebaut (z.B GCI Scripts, PHP – Scripts etc). Durch Dynamic Shared Objects ist es möglich Module dynamisch in den Apache einzubinden oder zu entfernen. Zur Laufzeit können sie verändert werden, ohne aktive Verbindungen unterbrechen zu müssen und ohne den Apache neu übersetzen zu müssen. Wenn der Apache statisch übersetzt werden soll, muss man sich vorher Gedanken über dessen Aufbau machen, welche Module benötigt werden. Bei Verwendung der DSO Funktionalität kann jedoch auch nachträglich die Modulzusammensetzung geändert werden.

Kapitel 7

Die Konfigurationsdatei des Apache heißt http.conf und ist in drei Bereiche aufgeteilt.

- globale Einstellungen (Arbeitsumgebung)
- Einstellungen des Hauptspeichers (Anweisungen zur Arbeitsweise)
- Einstellungen für den virtuellen Server.

Dokumentenverzeichnis = Wurzelverzeichnis für alle veröffentliche Dokumente.
DocumentRoot definiert das Verzeichnis, das die HTML Dokumente und sonstige Dateien enthält, die abrufbar sind.

Kapitel 8

<u>.htaccess = Verzeichniskonfigurationsdatei</u>
Üblicher Weg, um nur bestimmten Benutzer den Zugriff auf bestimmte Daten zu erlauben. Verzeichnisse, Unterverzeichnisse, bestimmte Dateien, oder Dateitypen können geschützt werden. Passwortschutz kann für einzelne Benutzer oder für ganze Benutzergruppen eingerichtet werden. Dazu wird dann allerdings eine zusätzliche Datei, in der die Benutzer und Passwörter stehen, benötigt. Eine `.htaccess` ist durch bestimmte Schlüsselwörter gekennzeichnet.
AuthType: Art der Authentifizierung
AuthName: Benutzer kann dadurch mitgeteilt werden, für welchen Bereich er sich mit User Name und Passwort authentifizieren soll.
AuthUserFile: Bei ~ wird die Datei angegeben, in der die authentifizierten Benutzer und ihre Passwörter stehen (durch absolute Pfadangabe, ab dem Wurzelverzeichnis des Server Rechners). Besser Datei außerhalb des Web Projekts abzustellen
AuthGroupFile: Wenn Gruppen angegeben werden möchten.
Require: `User` oder `group` gibt an, ob Benutzer oder Gruppen gemeint sind. Sollen alle in der Passwortdatei aufgeführten Benutzer Zugriff auf das geschützte Verzeichnis gewährt werden, dann muss als 2. Schlüsselwort `valid-users` stehen.

Entsprechend muss eine Datei mit User und ihren Passwörtern angelegt werden. Sie kann mit Hilfe des Befehls `htpasswd` geschehen.
Datei wird mit `htpasswd [Optionen]` Passwortdatei `user [Passwort]` erstellt. Beim Anlegen der Datei wird die Option `-c` angegeben, danach nicht mehr.
Gruppendatei besteht aus Einträgen, bei denen zunächst ein Gruppenname notiert wird und dahinter die Namen von Benutzer, die zu dieser Gruppe gehören.

Schutz auf Dateien/Dateitypen/Zugriffsmethoden

```
<Files *.html>
     require user  ela, jochen
</Files>
```
= Schutz auf html Dateien beschränkt!

Schutzmechanismen, die mit Hilfe von `.htaccess` erstellt wurden, sind auf http Ebene sicherer als CGI Skripte. Allerdings `.htaccess` kein Generalschutz. Wenn mit einem anderen Internet Protokoll zugegriffen wird, gilt der Schutz nicht.
Es ist auch möglich IPs/IP Bereiche oder Namensadressen zuzulassen/auszuschließen. Nicht authentifizierte Benutzer erhalten dann eine Fehlermeldung. Sinnvoll, wenn z.B. nur Mitarbeiter das Web Angebot zugänglich gemacht werden soll.

Kapitel 9

Kommunikation zwischen Client A und Client B ist öffentlich, Problem des sicheren Datenaustausches. Aktive und passive Angriffe vom Man in the middle. Lösung Ausweis, mit sich A gegenüber B authentifizieren kann.

Symmetrische Verschlüsselung:
Ein Schlüssel zum Ver -, und Endschlüsseln der Nachricht. Der Schlüssel muss allen Beteiligten bekannt sein. Es können große Datenmengen in kurzer Zeit ver -, und entschlüsselt werden.

Asymmetrische Verschlüsselung
Sender und Empfänger verwenden unterschiedliche Schlüssel zur Ver, -und Entschlüsselung. Sog. Schlüsselpaar `public` und `private key`. Daten, die mit dem `private Key` verschlüsselt wurden, können nur mit dem `public` entschlüsselt werden und umgekehrt. Empfänger erzeugt das Schlüsselpaar und übermittelt den `public key` dem Sender. Dieser verwendet ihn um seine Daten zu verschlüsseln. Entschlüsselt können die Daten allerdings nur vom Empfänger mit dem `private key`.

Secure Socket Layer SSL
Um sicheren Webserver einzurichten, sollte ein Webserver der SSL unterstützt, verwende werden. Das SSL Protokoll unterstützt die symmetrische als auch die asymmetrische Verschlüsselung. Asymmetrische zu Beginn des Verbindungsaufbaus und zur Vereinbarung des symmetrischen Schlüssels.
SSL ist ein Protokoll für sichere, verschlüssele Übertragung von Nachrichten im Internet. Für beliebige Dienste wie z.B. SMTP, FTP, POP3.

Verbindung über SSL
Der Aufbau einer gesicherten Website erfolgt durch die Angabe des Schemas https in der URL. Zu Beginn stellt der Client eine Anfrage an den Server und schickt ihm die Verschlüsselungsverfahren, die er unterstützt. Der Server wählt ein Verfahren aus und übermittelt dem Client sein Zertifikat mit dem öffentlichen Schlüssel des Servers. Anschließend generiert der Client den sog. Sitzungsschlüssel für ein sym. Verschlüsselungsverfahren. Dieser Session Key wird nun mit dem öffentlichen Schlüssel des Servers verschlüsselt zum Server übertragen. Nur dieser kann ihn nun mit dem privaten Schlüssel wieder entschlüsseln.

Host und Client – Authentisierung
Verschlüsselung alleine reicht nicht aus, um eine Verbindung sicher zu machen.
Der Server muss noch einen Nachweis über seine Identität bringen => Host-Authentisierung
Wenn sich auch der Benutzer ausweisen muss, spricht man von einer Client-Authentisierung

x.509 Zertifikate

Die Garantie, dass ein öffentlicher Schlüssel einer bestimmten Person oder Entität gehört, wird im Internet durch ein Zertifikat gewährleistet. Der Internet Standard X.509 beschreibt die Form eines Software-Zertifikates. Es beinhaltet den Namen, die digitale Signatur des Ausstellers, sowie Informationen über die Identität des Inhabers. Das Zertifikat wird von einer Zertifizierungsstelle ausgestellt. Sie verbürgt sich dafür, dass die Zuordnung eines Public Keys zu einer bestimmten Institution korrekt ist und die Institution, die den passenden privaten Schlüssel besitzt, tatsächlich existiert.

Erstellen von Zertifikaten

Das Erstellen von Zertifikaten lässt sich grob in 3.Schritte gliedern. 1.ein Schlüsselpaar wird erstellt, 2.das Paar wird zusammen mit anderen Informationen (Name und andere Parameter des Schlüsselinhabers) als Certificate Signing Request (CSR) an die Zertifizierungsinstanz geschickt. Sie prüft mit geeigneten Maßnahmen die Richtigkeit dieser Angaben und den CSR mit der Signatur der CA zu bestätigen, sozusagen zu beglaubigen, und als fertiges Zertifikat zurück an den Antragssteller zuschicken.

Common Gateway Interface (CGI)

CGI ist eine Schnittstelle des Web – Servers, die es erlaubt Anfragen eines Web – Browsers an Programme auf dem Web – Server weiterzureichen und von diesen ausführen zu lassen. Die Programme können in verschiedenen Programmiersprachen wie z.B. C, C++, Java oder in speziellen Skriptsprachen wie Perl oder PHP geschrieben sein. Solche Programme können z.B Formulareingaben aus HTML Dateien verarbeiten, auf dem Server Daten speichern und dort gespeicherte Daten auslesen. Web Seiten werden zu Oberflächen für Anwendungen (elektronische Warenbestellung oder zum Abfragen von DB). Die CGI Schnittstelle ruft nicht nur ausführende Programme auf und leitet dessen Antwort weiter, sondern stellt auch eine Reihe von Daten bereit, die der Web – Server speichert und die ein CGI Skript auslesen kann, um Daten verarbeiten zu können. Diese Daten werden in sog. CGI Umgebungsvariablen gespeichert. Vorteil der CGI Schnittstelle ist, dass es sich um einen kommerziell unabhängigen, kostenlosen und produktübergreifenden Standard handelt.

Log- Dateien

Um einen Überblick über die Funktion und Auslastung des Servers zu bekommen, kann der Apache verschiedene Logfiles führen. Ein Web – Server protokolliert jeden an ihn gerichteten Request in einer `AccessLog` Datei. Falls bei der Bearbeitung Fehler auftreten, werden diese in der `ErrorLog` Datei registriert. Für virtuelle Server existieren eigene `AccessLog` Dateien => geringerer Verwaltungsaufwand. `ErrorLog` in der `http.conf` gibt an, in welche Datei er seine Fehlermeldungen schreiben soll. Eintragungen enthalten Informationen über Fehler, die während es Betriebs des Servers und beim Bearbeiten und beim Beantworten von Anfragen erfolgen.