Tunneling

Einsatz von Tunnels für die Umstellung von IPv4 auf IPv6

Wolfgang Schulte

Die breite Einführung des Internet Protocol Version 6 (IPv6) als Basisprotokoll der Internetschicht ist früher oder später in den Netzen zu erwarten. Die Zeit der Freaks und Akademiker, die IPv6 als Testumgebung einsetzen, ist sicher bald vorbei.
Windows XP, Vista und Windows 7 bieten standardmäßig die Unterstützung für IPv6 an. Dabei wurden für die vollständige und problemlose Umstellung verschiedene Stategien vorgeschlagen. Eine Möglichkeit ist der Einsatz von Tunnels bzw. der duale Protokoll-Stack sowohl für IPv4 als auch IPv6.

Wolfgang Schulte ist Dozent an der Dualen Hochschule Baden-Württemberg Stuttgart

Mit Tunneling oder Encapsulation wird der Prozess bezeichnet, bei dem zwei unterschiedliche Protokolle auf der gleichen Schicht paarweise zusammen übertragen werden. Die Daten des einen Protokolls werden in die Datenpakete des zweiten Protokolls eingepackt. Dazu definiert das RFC 1983 Internet Users' Glossary: „Tunnelling refers to encapsulation of protocol A within protocol B, such that A treats B as though it were a datalink layer. Tunneling is used to get data between administrative domains which use a protocol that is not supported by the internet connecting those domains."

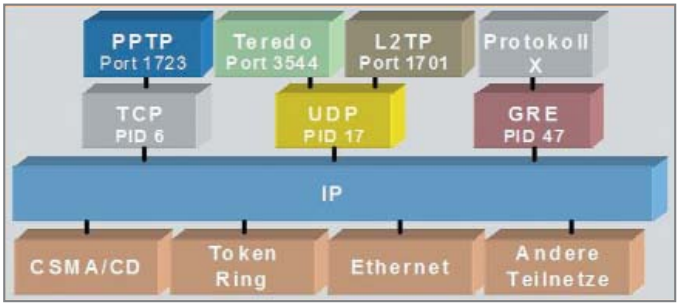


Bild 1: Protokollstapel für einige Tunneling-Protokolle

Einige der Tunneling-Protokolle sind:

- GRE RFC 2784: Einkapselung eines Protokolls auf der Internetschicht;
- RFC 2637: Point-to-Point Tunneling Protocol (PPTP);
- RFC 2661: Layer Two Tunneling Protocol (L2TP);
- RFC 4380: IPv4 Host hinter NAT über IPv6 Teredo;
- WinScP.

In *Bild 1* ist der Protokollstapel für einige Tunneling-Protokolle dargestellt. Die entsprechende IP-Identifikation bzw. die UDP- und TCP-Ports sind darin ausgewiesen.

Generic Routing Encapsulation (GRE) setzt direkt auf der Internetschicht auf IP auf. Teredo läuft als Anwendung, wie auch L2TP, über dem User Datagram Protocol (UDP). Das Point-to-Point Tunneling Protocol (PPTP) liegt als Anwendung über dem Transport Control Protocol (TCP) auf Schicht 4.

Generic Routing Encapsulation

RFC 2784 GRE ist ein von Cisco entwickeltes Protokoll und von der Internet Society erstellter RFC (Requests for Comments) für die Verkapselung von einem beliebigen Protokoll (z.B. IPv6 oder IPX) auf der Internetschicht in ein beliebiges anderes Protokoll (z.B. IPv4) auf der gleichen Schicht. Es setzt direkt, wie TCP und UDP, auf IP auf und verwendet die IP-ID 47. Dabei müssen sowohl der Anfangs- als auch der Endpunkt des Tunnels von einem GRE-fähigen Gerät gebildet werden.

Im GRE-Frame-Format (*Bild 2*) zeigt das C-Flag im GRE-Header die Gültigkeit der Prüfsumme (Checksum) an. Die Version (3 bit) ist 0. Das Feld Protokolltyp enthält den Protokolltyp des folgenden Nutzlastpakets. Die Protokolltypen sind im Sinne von RFC1700 bzw. bei IANA als Ethernet Type definiert.

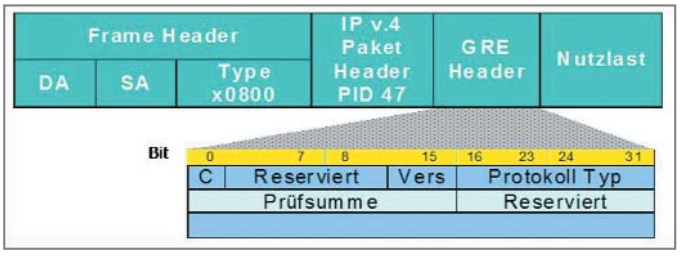


Bild 2: Frame-Format für GRE

Beispiel: Wenn IPv4 als Nutzlast übertragen wird (IPv4 über IPv4), enthält das Feld Protokolltyp x0800.

In *Bild 3* ist ein Tunnel mittels GRE von Router 1 nach Router 2 eingerichtet

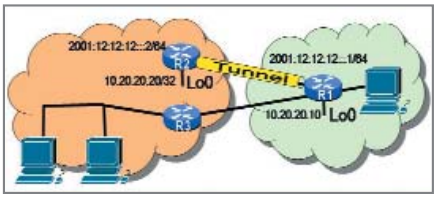


Bild 3: Tunneling zwischen zwei Netzen

worden. In Router 2 ist dieser Tunnel ebenfalls zu konfigurieren.

```
R1
interface Tunnel12
 no ip address
 ipv6 address 2001:12:12:12::1/64
 tunnel source Loopback0
 tunnel destination 10.20.20.20
 tunnel mode gre
```

In obigem Beispiel wird IPv6 über IPv4 getunnelt.

Teredo

Falls ein Host mit einer privaten IP-Adresse hinter einer Network Address Translation (NAT) in einem privaten Netz nach RFC 1631 angebunden ist, ist die Verbindung zu einem Host in einem öffentlichen IPv6-Netz u.a. durch einen Tunnel herzustellen. Dieser kann z.B. mit Teredo RFC 4380 Tunneling IPv6 over UDP through Network Address Translations bereitgestellt werden. Der RFC 4380 stellt den Hosts einen Service zur Verfügung, der hinter einer IPv4-NAT die Verbindung zu IPv6 mittels UDP ermöglicht.

Ein Teredo-Tunnel (T-Tunnel) besteht aus folgenden Komponenten (*Bild 4*):

- Die Teredo-Clients (T-Clients) hinter der NAT im privaten Netz sollen eine Verbindung in das IPv6-Netz zum Host C bekommen.

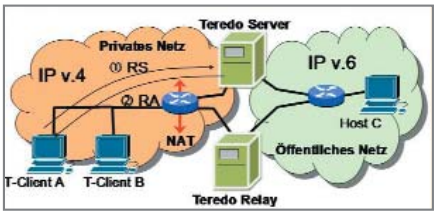


Bild 4: Komponenten eines Teredo-Tunnels

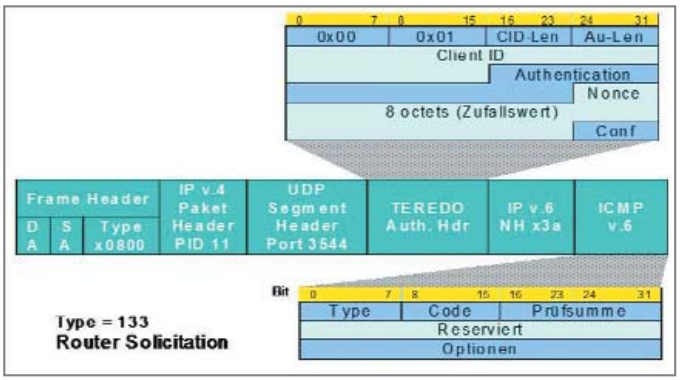
- Der Teredo-Server (T-Server), ein IPv4/v6-Gerät, stellt die Information zur Bildung einer T-Adresse bereit und kennt das Teredo-Relay.
- Das Teredo-Relay (T-Relay), ein IPv4/v6-Gerät, stellt den Tunnel zwischen T-Clients und dem IPv6-Host C her.

Der Aufbau des Tunnels geschieht in drei Schritten:

- Konfigurationsprozess zur Beschaffung der Information zwecks Bildung einer Teredo-Adresse (T-Adresse) für den T-Client;
- Erstellung der speziellen Teredo-Adresse im T-Client;
- Einleitung der Kommunikation vom T-Client durch den T-Tunnel über das T-Relay.

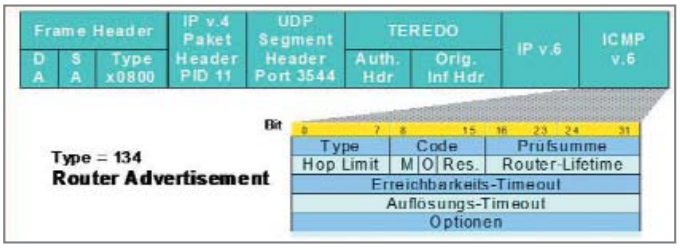
Der Konfigurationsprozess bei Cone-NAT startet wie folgt:

Bild 5: Router-Anfrage Nachricht



- Der T-Client sendet einen IPv4/UDP-Rahmen mit Teredo Authentication Header, mit IPv6 und einer ICMPv6-RS-Nachricht (RS – Route Solicitation) an den T-Server (*Bild 5*). Diese Anforderungsnachricht sendet der T-Client mit seiner IPv4-Adresse im privaten Netz hinter der NAT. In ihr ist ein Flag (Cone-Flag) gesetzt, das anzeigt, welche Eigenschaften die NAT (Cone-NAT) hat. Diese Router Solicitation wird als IP-Paket mit UDP-Segment-Port 3544 Teredo an den T-Server gesendet.
- Der T-Server antwortet mit einer RA-Nachricht (RA – Router Advertisement), *Bild 6*. Da der T-Client in der RS-Nachricht das Cone-Flag gesetzt hatte, schickt der T-Server die RA-Nachricht zu einer zweiten, alternativen IPv4-Adresse. Erhält der T-Client diese RA-Nachricht, ist er sicher hinter einer Cone-NAT. Mit der erhaltenen Information im Originator Header kann der T-Client jetzt die besondere Teredo-Adresse erstellen.

Bild 6: Router-Anzeige Nachricht



Adressformat bei Teredo

Für die Aufgaben und Bildung des Tunnels wird für Teredo ein eigenes Adressformat spezifiziert (*Bild 7*):

- Der Teredo Präfix ist bei IANA mit 2001:0000::/32 Teredo festgelegt.
- Es folgt die hexadezimale IPv4-Adresse des T-Servers.
- In den Flags stehen die Art der NAT (z.B. Cone-NAT) und eine 12-bit-Zufallszahl.
- Die nächsten 16 bit enthalten die externe Portnummer des T-Clients.
- Es folgt die externe IPv4-Adresse des T-Clients.

Die Port-Nr. und die IPv4-Adresse des T-Clients werden zur Sicherheit mit xff XORed gesetzt, damit die NAT diese Adressteile nicht umsetzt.

Bei den Verfahren zur NAT werden im RFC 3489 – STUN – Simple Traversal of

2001:0:	d5c7:a2d6	0x8000 Cone 0x0 kein Cone	1b81	f5f5:213b
Teredo Präfix	IP v.4 Adr T-Server	Flags (Nat-Type	Externer Port	Externe IP v.4 Adresse
Bit 32	32	16	16	32

Bild 7: Teredo-Adresse

User Datagram Protocol (UDP) folgende vier NAT-Varianten unterschieden:

- Cone-NAT: Eine abgehende Information von einem internen Rechner mit privater IP-Adresse und Port-Nr. (192.168.2.100:2345) über die NAT erlaubt nach der NAT-Adressübersetzung (private zu öffentliche Adresse) anschließend die Kommunikation mit externen Rechnern. Externe Rechner können über die externe Adresse der NAT-Verbindung zum internen Rechner eine Verbindung aufbauen.
- Restricted Cone-NAT: Externe Rechner können den internen Rechner nur dann erreichen, wenn der interne Rechner vorher diesen externen Rechner kontaktiert hatte.
- Port Restricted NAT: Die Port Restricted NAT arbeitet wie die Restricted NAT, wenn die externe Kontaktaufnahme über den gleichen Port erfolgt.
- Symmetric NAT: Hier ist die Verbindung zum internen Rechner durch externe Hosts nicht oder nur schwer möglich. Wenn der interne Rechner mit seiner privaten IP-Adresse und seinem Port zu einem anderen externen Rechner als zuvor überträgt, wird diese neue Verbindung mit einem anderen Mapping abgelegt.

Durch entsprechenden Austausch der Teredo-Rahmen vom T-Client mit einem oder mehreren T-Servern kann die NAT-Variante festgestellt werden.

Der Kommunikationsprozess bei Cone-NAT startet wie folgt (*Bild 8*):

- Der T-Client muss zunächst die IPv4-Adresse des T-Relays ermitteln. Dazu sendet er eine IPv4/UDP-Nachricht mit IPv6- und ICMPv6-Echo-Request-Nachricht (*Bild 9*) über den T-Server an Host C.
- Der T-Server leitet die ICMPv6-Echo Request-Nachricht an den IPv6-Host C weiter.
- Der IPv6-Host C antwortet mit einem ICMPv6-Echo-Reply an den nächsten T-Relay.
- Der T-Relay packt um den IPv6/ICMPv6-Rahmen wieder den Frame Header mit IPv4 und UDP und leitet diesen Reply dann an den anfragenden T-Client weiter.
- Der T-Client kennt jetzt den Weg über das T-Relay zum Host C; der Tunnel ist für weiteren Datenverkehr geöffnet.

Für eine Kommunikation zwischen zwei T-Clients hinter der gleichen NAT wird wie folgt vorgegangen (*Bild 10*):

- Der T-Client A sendet ein besonderes Paket (Bubble-Paket) an die Multicast-Adresse 224.0.0.253 (Teredo). Ein Teredo-Bubble-Paket wird in der Regel gesendet, um ein NAT-Mapping aufrechtzuerhalten und besteht aus einem IPv4-, UDP- und IPv6-Header ohne IPv6-Nutzinformation.
- Nach Empfang der Nachricht von T-Client A speichert T-Client B die erhaltene Adresse und sendet eine Unicast-Nachricht (Bubble-Paket) an T-Client A zurück. T-Client B erkennt jetzt, dass T-Client A ebenfalls intern (hinter der NAT) im gleichen Netz liegt.
- Die beiden T-Clients können jetzt mit dem Datenverkehr beginnen.

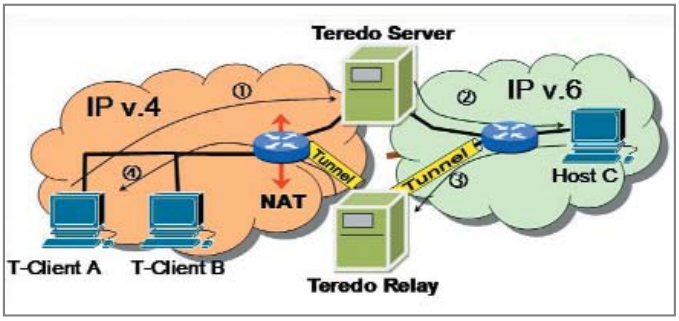


Bild 8: Bildung des Teredo-Tunnels

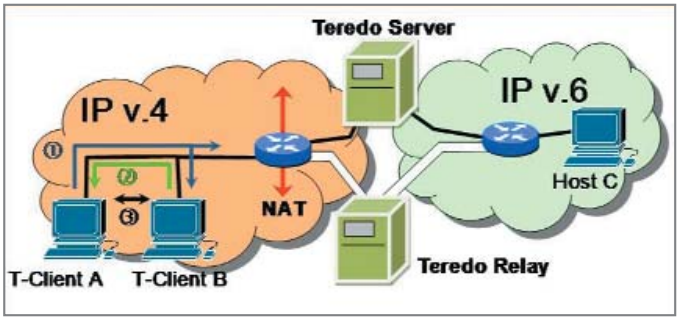


Bild 10: Kommunikation der T-Clients

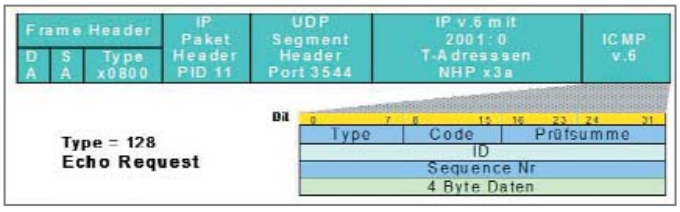


Bild 9: Format der Echo-Request-Nachricht

Point-to-Point Tunneling Protocol

Das RFC 2367 PPTP wurde 1996 vom PPTP-Forum entwickelt (*Bild 11*). Es kommt hauptsächlich in Microsoft-Betriebssystemen zum Einsatz und ist weitgehend von RFC 2661 L2TP abgelöst worden. Es stellt eine Erweiterung des Point-to-Point-Protokolls (PPP) dar, und zwar wird das PPP durch ein IP-Netz getunnelt und bietet darüber hinaus Verfahren zur Authentifizierung, Komprimierung und Verschlüsselung an.

PPTP ist ausschließlich für die Übertragung von IP, IPX und NetBEUI über IP vorgesehen.

PPTP-Architektur

Die PPTP-Architektur kennt zwei logische Komponenten – den PPTP Access Concentrator (PAC) und den PPTP Network Server (PNS). Der PAC verwaltet die Verbindungen und stellt diese zum PNS her. Der PNS ist für das Routing und die Kontrolle der vom PNS empfangenen Pakete zuständig.

Der PAC ist üblicherweise in den Client integriert und stellt eine PPP-Verbindung zum PNS her. Nach der Authentifizierung und Autorisierung wird dem PAC eine IP-Adresse aus dem LAN zugewiesen. Danach beginnt er, PPTP-Pakete zu senden. Die PPP-Rahmen werden mit Generic Routing Encapsulation (GRE) verpackt. Danach werden die Datenpakete über das IP-Netz zum Ziel transportiert.

Eine Verschlüsselung findet nicht statt. Daher muss bereits bei PPP die Verschlüsselung ausgehandelt werden.

PPTP-Funktion

Typischerweise läuft eine Kommunikation über PPTP folgendermaßen ab:

- Zunächst wird mittels TCP eine sichere PPTP-Verbindung vom PAC zum PNS aufgebaut.
- Anschließend startet das PPTP mit den Nachrichten für die Control-Verbindung, gefolgt von den Nachrichten für das Call-Management. Den Abschluss der PPTP-Nachrichten beim Verbindungsaufbau bildet die Nachricht für die Session Control. Über den TCP-Port 1723 laufen alle PPTP-Kontrollnachrichten einer PPTP-Verbindung. Dieser Port muss bei der Nutzung von PPTP geöffnet sein, damit ein PPTP-Client die ausgehenden bzw. eingehenden Verbindungen nutzen kann.
  Der Start-Control-Connection-Request ist eine PPTP-Control-Nachricht zur Herstellung der Control-Verbindung zwischen PNS und PAC.
- Zum Aufrechterhalten der Verbindung werden PPTP-Echo-Request- und PPTP-Reply-Nachrichten ausgetauscht.
- Der Datenverkehr wird dann über IP, GRE, PPP, IP, TCP und Datenrahmen im Tunnel abgewickelt.
- Der Verbindungsabbau geschieht mittels PPTP-Clear-Call-Request-, Call-Disconnect-Notify- und Stop-Session-Request/Reply-Nachrichtenaustausch.

Layer-2-Tunneling Protocol

Mit dem RFC 2661 Layer-2-Tunneling Protocol (L2TP) wurde die Aufgabe der Erstellung einer PPP-Verbindung über ein IP-Netz zwischen zwei Netzstationen gegenüber dem PPTP erweitert. Nicht nur IP, IPX und NetBEUI, sondern jedes beliebige Protokoll sollte jetzt den Tunnel benutzen können. Anstelle einer reellen Punkt-zu-Punkt-Verbindung über z.B. PSTN oder ISDN besteht die Übertragungsstrecke aus mehreren Routern, die miteinander verbunden sind. Für dieses Szenario gibt es zwei Protokolle:

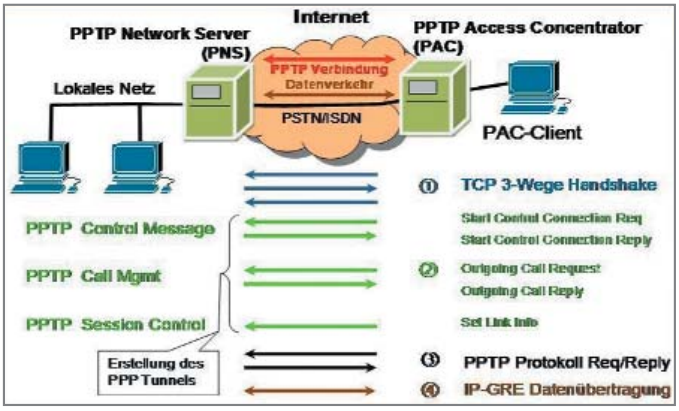
- L2F: Layer-2-Forwarding (RFC 2341);
- PPTP: Point-to-Point Tunneling Protocol (RFC 2367).

Sie bilden die Basis für das Layer-2-Tunneling Protocol. L2TP bietet selbst keinen Authentifizierungs-, Integritäts- und Verschlüsselungsmechanismus. Ein Schutz der zu übertragenden, getunnelten Daten kann z.B. mit IPsec erfolgen.

L2TP-Architektur

Die L2TP-Architektur definiert mit L2TP Access Concentrator (LAC) und L2TP Network Server (LNS) zwei logische Systeme (*Bild 12*). Der LAC verwaltet die Verbindungen und stellt diese zum LNS her. Der LNS ist für das Routing und die Kontrolle der vom LAC empfangenen Pakete zuständig. Das L2TP definiert die Kontroll- und Datenpakete zur Kommunikation zwischen LAC und LNS. Ein Network Access Server (NAS) stellt einen temporären Zugang für Remote-Systeme zur Verfügung. Der NAS kann alternativ im LAC oder im LNS implementiert sein.

Es gibt insgesamt zwei Szenarien, um einen L2TP-Tunnel aufzubauen. Das erste sieht eine PPP-Verbindung zwischen dem Client und dem LAC vor, beispielsweise PSTN oder ISDN. Der LAC tunnelt die PPP-Daten zum LNS und bekommt von diesem eine IP-Adresse aus dem LAN zugeteilt. Das zweite Szenario sieht eine direkte Unterstützung von L2TP auf dem LAC-Client vor, der dann selbst der LAC ist. Die Daten werden genauso mit PPP übertragen. Die IP-Adresse aus dem LAN wird auch hier vom LNS zugeteilt.



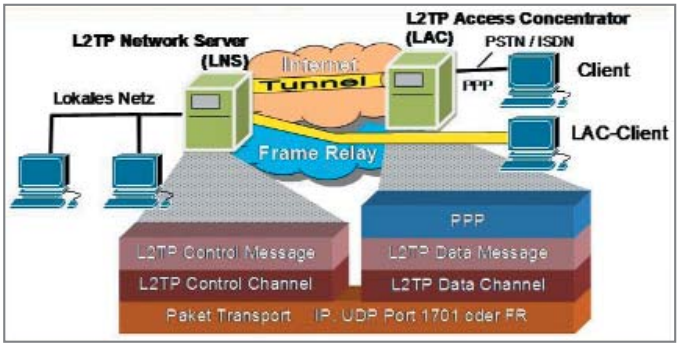


Bild 11 (links): PPTP-Kommunikation
Bild 12 (oben): L2TP-Kommunikation

In beiden Fällen ist die Autorisierung und Authentifizierung von den Mechanismen im LAN abhängig.
Mit L2TP wird ein Tunnel zwischen LAC und LNS aufgebaut. Der NAS identifiziert den Remote-User über einen Authentifizierungsserver. Ist die Authentifizierung erfolgreich, wird der L2TP-Tunnel etabliert. Der LNS identifiziert sich ebenfalls beim Remote-User und bestätigt den L2TP-Tunnel. In diesem Tunnel wird für jede PPP-Verbindung eine Sitzung (Session) zwischen LAC und LNS aufgebaut. Mit Hilfe des Multiplexmodus lassen sich in einem Tunnel mehrere Sitzungen aufbauen.
Innerhalb des PPP-Tunnels existieren zwei verschiedene Kanäle. In einem werden die Kontrollnachrichten übertragen, in dem anderen die eigentlichen Nutzdaten. Der Kontrollkanal ist eine sichere Verbindung über TCP, der Datenkanal ist eine ungesicherte Verbindung mit UDP-Port 1701. Die Nutzdaten werden also ungesichert in Klartext übertragen, sofern das Transportprotokoll (PPP) keine Verschlüsselung unterstützt oder nicht aktiviert wurde. *(bk)*