

Das Kerberos-Protokoll

Florian Wallner
wallner@balumba.org

Juli 2008

Das Kerberos-Protokoll ermöglicht das sichere Authentifizieren in unsicheren Computernetzwerken. Es wurde in den 80'er Jahren des 20. Jahrhunderts von Steve Miller und Clifford Neumann im Rahmen des Athena-Projekts am MIT entwickelt.

1 Hintergrund und Geschichte

Anfang der 80'er Jahre zeichnete sich in der EDV ein Paradigmenwechsel ab: Traditioneller Weise waren Computer große, zentral gewartete Systeme, an welche die Nutzer über einfache, serielle Terminals angeschlossen waren und deren zur Verfügung stehende Ressourcen zwischen allen Nutzern mehr oder weniger gerecht verteilt wurden. Diesen zentralen Großrechnern wurde durch die Verfügbarkeit von kleinen, preiswerten Computern zunehmend Konkurrenz gemacht. Es wurde mit einem Mal üblich, dass Nutzer eigene, wenn auch leistungsschwächere, Computer auf ihren Schreibtischen stehen hatten, die über ein Netzwerk untereinander und mit einigen Servern verbunden waren.

Die Erkenntnis, dass den Computern der Nutzer und dem sie verbindenden Netzwerk nicht mehr zu trauen war, da der Nutzer unter Umständen in der Lage ist, jede Software-Komponente seines Computers, bis hin zum Betriebssystem, auszutauschen, führte zur Entwicklung des Kerberos-Protokolls im Rahmen des Athena-Projekts am MIT.

1.1 Das Needham-Schroeder-Protokoll

Als Grundlage für die Entwicklung des Kerberos-Protokolls diente das 1978 von Roger Needham und Michael Schroeder am Xerox Parc entwickelte Needham-Schroeder-Protokoll.

Kerberos unterscheidet sich jedoch in zwei Punkten maßgeblich vom Protokoll von Needham und Schroeder:

- Der Verkehr im Netzwerk wird deutlich reduziert. Dafür ist das Kerberos-Protokoll von synchronisierten Uhren aller teilnehmenden Systeme abhängig.
- Kerberos ermöglicht, im Gegensatz zum Needham-Schroeder-Protokoll, ein *Single-Sign-On* des Nutzers. Das bedeutet: Er muss nur einmal am Tag sein Passwort eingeben. Dies war eine grundlegende Forderung an das Kerberos-Protokoll.

2 Das Protokoll

Jason Garman [Garman03] definiert Kerberos als

A secure, single-sign-on, trusted third-party, mutual authentication service

Diese Definition ist zwar gelungen, bedarf jedoch einer genaueren Erklärung:

Secure Dem Netzwerk ist nicht zu trauen. Ein potenzieller Angreifer ist in der Lage, den Datenverkehr mitzulesen und zu verändern oder selbst Daten ins Netzwerk einzuspeisen. Von daher werden Passwörter nie über das Netzwerk übertragen. Weder im Klartext noch verschlüsselt.

Single-Sign-On Der Benutzer muss nur einmal am Tag sein Passwort eingeben. Danach ist er in der Lage, alle angebunden Dienste ohne weitere Eingabe seines Passworts zu benutzen.

Trusted Third-Party Im Kerberos-Protokoll existiert eine zentrale Instanz der von allen beteiligten Parteien vertraut wird. Dieser zentralen Instanz sind alle in der Installation verwendeten kryptographischen Schlüssel bekannt.

Mutual Gegenseitige Authentifizierung. Kerberos gewährleistet nicht nur eine Authentifizierung des Nutzers gegenüber dem Dienst, den er benutzen möchte, sondern stellt sicher, dass der Dienst wirklich der ist den der Nutzer erwartet.

Das Kerberos-Protokoll ist momentan in zwei Versionen verbreitet Version Vier und Version Fünf (Version Eins bis Drei wurden ausschließlich intern am MIT verwendet). Das Protokoll in Version Vier ist nur durch den Quelltext der vom MIT veröffentlichten Software spezifiziert, während der Nachfolger, Version Fünf in mehreren RFCs der IETF spezifiziert wurde. Aktuell definieren die RFCs mit den Nummern 4120 und 4121 das Protokoll.

Es gibt Kerberos-Implementierungen für alle gängigen Betriebssysteme, in den moderneren Varianten werden sie von Hause aus mitgeliefert. Am weitesten verbreitet dürfte mit Abstand die Implementierung von Microsoft sein. Kerberos ist ein integraler Bestandteil von Microsofts *Active Directory Service*.

Weiterhin gibt es noch die Referenzimplementierung des MIT sowie das in Schweden entwickelte Heimdal-Kerberos.

2.1 Realms und Principals

Als *Realm* bezeichnet man im Kerberos-Protokoll eine Installation, die von allen anderen Installationen verschieden ist und unter einer administrativen Kontrolle stehen. Die Realm wird üblicherweise durch den Domain-Namen in Großbuchstaben bezeichnet, zum Beispiel BALUMBA.ORG. Hierbei handelt es sich jedoch nur um eine Konvention. Der Name einer Realm unterscheidet zwischen Groß- und Kleinschreibung, BALUMBA.ORG ist eine andere Realm als balumba.org.

In einer Kerberos-Installation, wird jede Einheit die dem System bekannt ist, als "*Principal*" bezeichnet. Principals können sowohl einzelne Benutzer, als auch Rechner oder auf Servern laufende Dienste sein. Dargestellt werden Principals im einfachsten Fall als $\langle \text{Benutzername} \rangle @ \langle \text{Realm} \rangle$ also zum Beispiel wallner@BALUMBA.ORG. Ist der Principal ein Dienst, wird er als $\langle \text{Dienstname} \rangle / \langle \text{Host-Name} \rangle @ \langle \text{Realm} \rangle$ dargestellt also zum Beispiel SMTP/mail.balumba.org@BALUMBA.ORG. Host-Namen werden immer voll ausgeschrieben, dadurch ist es möglich mehrere Rechner mit gleichen Host-Namen in unterschiedlichen Domains zu verwalten.

2.2 Das Key-Distribution-Center

Das zentrale Element einer Kerberos-Realm ist das *Key-Distribution-Center* (kurz: KDC). Es besteht aus drei Teilen:

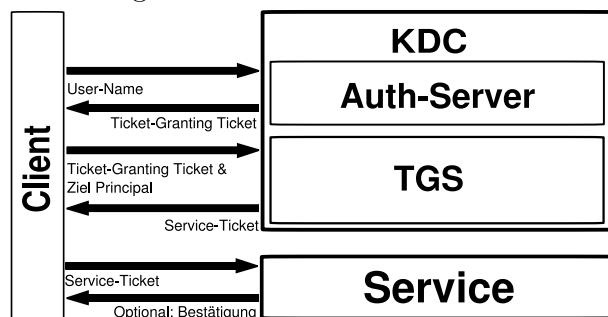
- Einer Datenbank aller in der Realm bekannten Principals und den ihnen zugeordneten kryptographischen Schlüsseln.
- Dem **Authentication-Server**. Dieser erteilt das Ticket-Granting-Ticket (TGT), mit welchem dann die Tickets für die einzelnen Dienste angefordert werden können.
- Dem **Ticket-Granting-Service** (kurz TGS): Von ihm bekommt der Nutzer nach Vorlage des Ticket-Granting-Tickets das Zugangsticket für die einzelnen Dienste.

Üblicherweise sind diese drei logisch voneinander unabhängigen Dienste als ein Programm implementiert und teilen sich einen Adressraum.

3 Ablauf

Im Folgenden wird der Ablauf einer Kerberos-Session genauer erklärt.

Abbildung 1: Der Ablauf einer Kerberos-Session



3.1 AS-REQ

Meldet sich der Nutzer zu Beginn seines Arbeitstages an seiner Workstation an, beantragt er Auth-Server das so genannte *Ticket-Granting-Ticket*. Dieser Antrag ist unverschlüsselt und enthält den Principal des Nutzers, die aktuelle Zeit auf dem Rechner des Nutzers und den Principal des Ticket-Granting-Service. Die Zeit ist in diesem Antrag übrigens nicht enthalten, um so genannte *“Replay-Attacken”* zu verhindern, schließlich ist der *“AS-REQ”* genannte Antrag unverschlüsselt, sondern um frühzeitig im Vorgang nicht synchrone Uhren in der Kerberos-Installation zu erkennen.

3.2 AS-REP

Ist dem KDC der beantragende Principal bekannt und die Differenz zwischen der lokalen Systemzeit und dem Zeitstempel weichen nicht zu sehr von einander ab (üblicherweise darf die Differenz nicht mehr als fünf Minuten betragen), generiert der Auth-Server einen Session-Schlüssel, den er sich für den Rest der Session¹ mit dem Nutzer teilt.

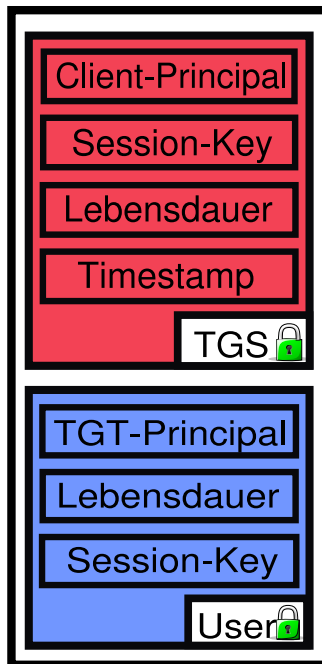
Dieser Session-Schlüssel wird dem Benutzer gemeinsam mit dem TGS-Principal und der Lebensdauer des im nächsten Absatz erklärten TGT mit seinem Schlüssel verschlüsselt zugesandt.

Außerdem enthält das AS-REP-Paket das so genannte *Ticket-Granting-Ticket* (kurz: TGT), bestehend aus dem Client-Principal, einer Kopie des erstellten Session-Schlüssels, der Lebensdauer und einen Zeitstempel. Das TGT ist mit dem Schlüssel des Ticket-Granting-Servers verschlüsselt und dadurch für den Nutzer nicht lesbar (siehe Abbildung 2).

Ist der Nutzer in der Lage, den für ihn bestimmten Teil des Pakets zu entschlüsseln, ist er im Besitz eines Session-Schlüssels, mit dem alle weitere Kommunikation gesichert wird. So wird sichergestellt, dass mit dem Langzeit-Schlüssel des Nutzers verschlüsselte Pakete nur selten über die Leitung gehen (ein- bis zwei mal pro Session). Dadurch wird das Risiko minimiert, dass ein

¹Eine Session dauert in der Regel 8 bis 24 Stunden, oder endet wenn der Benutzer sein Ticket-Granting-Ticket zerstört (z.B. weil er sich ausloggt)

Abbildung 2: Das AS-REP Paket



Angreifer sich durch lauschen auf der Leitung in Besitz eines solchen Paketes zum durchführen einer Offline-Attacke bringt, und damit den Schlüssel des Nutzers bricht.

3.3 Pre-Authentication

Das Kerberos-Protokoll in der Version 4 hat auf einen **AS-REQ** jedem Client ein TGT für jeden gewünschten Principal ausgestellt. Dies ermöglichte potenzielle Angreifern, sich in Besitz von Paketen, die mit dem Schlüssel eines beliebigen Principals verschlüsselt waren zu bringen und diesen Schlüssel mit einer Offline-Attacke zu brechen.

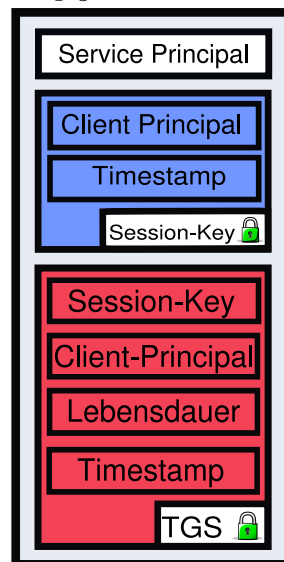
Diese Schwachstelle wurde in der Version 5 des Protokolls mit der Einführung der sogenannten *Pre-Authentication* (kurz: Pre-Auth) Phase beseitigt. Pre-Auth verlangt, dass der Antragsteller sich Authentifiziert, bevor das KDC für einen Principal ein Ticket ausstellt.

Wenn der Client eines Nutzer ein initiales TGT beantragt, teilt ihm ein entsprechend konfigurierter Auth-Server mit, dass er eine Authentifizierung des Nutzers verlangt, bevor er ihm das TGT ausstellt. Daraufhin sendet der Client das in Abschnitt 3.1 beschriebene AS-REQ Paket erneut, hängt aber einen mit seinem Schlüssel verschlüsselten Zeitstempel an. Da dem Auth-Server der Schlüssel des Nutzers bekannt ist, kann er den Zeitstempel entschlüsseln und somit den Nutzer authentifizieren.

3.4 TGS-REQ

Möchte der Nutzer im Laufe seiner Sitzung auf einen Kerberos-Fähigen Dienst wie zum Beispiel einen IMAP-, oder ein FTP-Server zugreifen, nutzt er das vom AS erhaltene Ticket-Granting-Ticket um beim TGS ein sogenanntes "*Service-Ticket*" zu beantragen. Diese Anfrage wird als "*TGS-REQ*" bezeichnet und ist in Abbildung 3 dargestellt.

Abbildung 3: Das TGS-REQ Paket



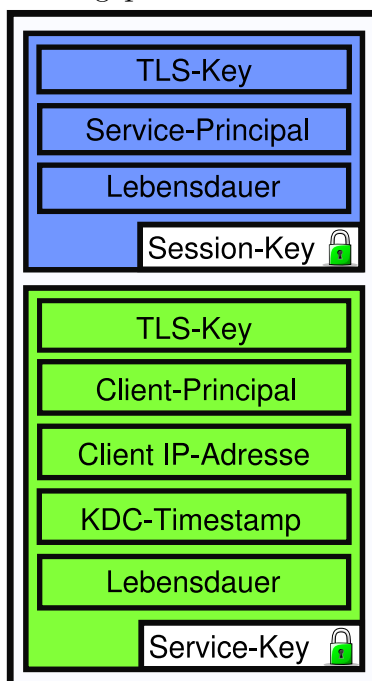
Die Anfrage besteht aus dem (unverschlüsselten) Principal des Dienstes für den das Service-Ticket beantragt wird, den Principal des beantragenden Clients und ein aktueller Zeitstempel, verschlüsselt mit dem im AS-REP erhaltenem Session-Key (der so genannte "*Authenticator*") und das ebenfalls ebenfalls mit dem AS-REP erhaltene Ticket-Granting-Ticket.

Durch den dem Antrag beigefügten Authenticator hat der beantragende Nutzer nachgewiesen, dass er im Besitz des vereinbarten Session-Schlüssels ist und somit der für den er sich ausgibt.

3.5 TGS-REP

Hat der Nutzer dem TGS gegenüber durch den *Authenticator* seine Identität nachgewiesen, stellt der TGS dem Nutzer ein Service-Ticket aus (dargestellt in Abbildung 4). Das Service-Ticket ist mit dem Schlüssel des Service' verschlüsselt (also für den Nutzer nicht lesbar) und enthält den Principals des Nutzers der den Antrag gestellt hat, die IP-Adresse von welcher der Antrag gestellt wurde, einen Zeitstempel mit der aktuellen Uhrzeit des KDCs und die Lebensdauer des Service-Tickets. Die Lebensdauer des Service-Tickets ist maximal so lang, wie das vom AS ausgestellte TGT noch Gültigkeit hat.

Abbildung 4: Das TGS-REP Paket



Zusätzlich zu diesen Informationen generiert das KDC noch einen weiteren Schlüssel, in Abbildung 4 als TLS-Key bezeichnet, der vom Dienst und Nutzer benutzt werden kann, um die Kommunikation untereinander zu verschlüsseln.

Außer dem Service-Ticket wird dieser Schlüssel auch noch zusammen mit dem Principals des Service, für den das Service-Ticket gültig ist, einem Zeitstempel und der Lebensdauer des Service-Tickets mit dem, dem Nutzer mit dem AS-REP übermittelten, Session-Schlüssel verschlüsselt im TGS-REP Paket zugeschickt.

3.6 AP-REQ und AP-REP

Der Nutzer ist nun in der Lage, sich mit der Vorlage des Service-Tickets beim Service für welchen dieses Ticket ausgestellt ist zu authentifizieren. Diese Schritte heißt AP-REQ und AP-REP, sollen aber an dieser Stelle nicht weiter beschrieben werden.

4 PKINIT

Das bis hierhin vorgestellte Verfahren nutzt ausschließlich symmetrische Kryptographie zur Sicherung des Protokolls. Dies bringt einige Nachteile mit sich:

- Die Provisionierung skaliert nicht: Jeder Teilnehmer an einer Realm muss beim Verwalter des KDC vorstellig werden um seinen Schlüssel zu hinterlegen, oder zu aktualisieren. Das mag bei einigen hundert Principals noch gehen, wird aber unter Umständen für eine Installation mit mehreren tausend Nutzern umständlich.
- Die Sicherheit des Systems wird letztendlich von der Stärke des Nutzerpassworts bestimmt. Eine alte Systemadministratorenweisheit sagt, dass die Nutzer prinzipiell schwache Passwörter wählen und die Aufklärung über die Gefahren solchen Verhaltens ein Kampf gegen Windmühlen ist.

Aus diesem Grund definiert das RFC 4556 Erweiterungen für Kerberos, die den initialen Schlüsselaustausch um asymmetrische Kryptographie erweitern.

Hier zeigt sich die Eleganz von Kerberos: Ein Eingriff war nur an einer Stelle nötig, um das Protokoll mit dieser Technik auszustatten.

Der Eingriff findet in den in den Abschnitten 3.1 und 3.2 beschriebenen Phasen AS-REQ und AS-REP statt und nutzt X.509-Zertifikate als Basis.

Mit dem AS-REQ-Paket schickt der Nutzer seinen öffentlichen Schlüssel und einen mit diesem Schlüssel signierten Zeitstempel. Diesen öffentlichen Schlüssel gleicht der AS mit seinen vertrauten Certification Authorities ab und wenn der Nutzer die richtigen Unterschriften auf seinem Schlüssel vorweisen kann und damit seine Berechtigung zur Authentifizierung nachgewiesen hat spezifiziert das RFC zwei Möglichkeiten für das weitere Vorgehen:

- Der AS schickt dem Nutzer wie gehabt einen symmetrischen Session-Schlüssel, den er mit seinem geheimen Schlüssel signiert und mit dem öffentlichen Schlüssel des Nutzers verschlüsselt zurück sendet.
- Der Nutzer und der AS handeln mit Hilfe des Diffie-Hellman Protokolls einen Session-Schlüssel aus, der im weiteren Verlauf der Session benutzt wird.

Wenn dem Nutzer das initiale Ticket-Granting-Ticket ausgestellt wurde läuft der Rest des Protokolls oben beschrieben weiter.

Es bleibt noch anzumerken, dass der einzige Hersteller, der im Augenblick PKINIT implementiert hat, Microsoft ist. Es ist jedoch davon auszugehen, dass es nur eine Frage der Zeit ist, bis diese Erweiterung auch in andere Implementationen Einzug findet.

5 Fazit

Mit dem Kerberos-Protokoll gelingt es, Nutzer über ein unsicheres Netzwerk zu authentifizieren. Es gibt jedoch ein paar Punkte, über die man sich beim Betrieb einer Realm im klaren sein muss:

- Auf dem KDC liegen alle in der Realm verwendeten Schlüssel im Klartext vor. Wenn ein Angreifer das System attackiert, ist also der KDC ein lohnendes Ziel und bedarf aus diesem Grund besonderer Aufmerksamkeit bei der Überwachung der Sicherheit des Systems. Dieser Punkt wird durch PKINIT adressiert.
- Das KDC muss aufgrund seiner Funktion immer erreichbar sein. Fällt es aus, geht das Authentifizieren in der gesamten Realm nicht mehr. Gültige Service-Tickets behalten natürlich ihre Gültigkeit.
- Kerberos erfordert in der gesamten Realm synchron laufende Systemuhren, denn durch den ständigen Umgang mit Zeitstempeln im Protokoll werden Replay-Attacken verhindert. Dafür müssen jedoch die Uhren synchron gehen.

Wenn man diese Punkte beachtet, ist ein Systemverwalter in der Lage, durch den Einsatz des Kerberos-Protokolls die Sicherheit eines Netzwerks deutlich zu erhöhen.

Literatur

[Garman03] Garman, Jason *Kerberos, The Definitive Guide*. Sebastopol 2003