Transbase®

WHITE PAPER

Transbase® Security

1. Überblick

Transbase® bietet verschiedenste Security Maßnahmen an, die sowohl die Authentifizierung der Benutzer als auch die physische Security der Speicher- und Kommunikationsmedien betreffen. Der Hauptspeicherinhalt eines laufenden Transbase® Prozesses (und auch des gemeinsamen Transbase® Shared Memories) ist nie verschlüsselt und nur - wie Files - durch Systemattribute geschützt.

Darüber hinaus ist das volle, durch den SQL-Standard definierte Privilegienkonzept implementiert (das hier nicht weiter besprochen wird).

Bevor die Mechanismen von Transbase® dargestellt werden, sollen einige grundlegende Begriffe aus der Kryptographie erläutert werden:

Ein **symmetrisches Kryptifizierungsverfahren** verwendet zur Verschlüsselung und zur Entschlüsselung den gleichen Schlüssel; der Schlüssel muss von beiden Seiten gekannt werden.

Ein **asymmetrisches Kryptifizierungsverfahren** verwendet zur Verschlüsselung einen anderen Schlüssel als zur Entschlüsselung; für jede Kommunikationsrichtung muss ein Schlüsselpaar existieren, wobei jede Seite nur jeweils einen der beiden Schlüssel kennen muss. Diese Verfahren werden häufig als Public-Key-Verfahren bezeichnet. Ihr Hauptnachteil ist die langsame Geschwindigkeit, weswegen sie im allgemeinen nur zum Austausch eines symmetrischen Schlüssels verwendet werden.

Als Kryptifizierungsverfahren werden je nach Anforderung typischerweise folgende eingesetzt: Unter einem "rotor-basierten Kryptifizierungsverfahren" versteht man im Prinzip eine zeichenweise Substitution, dessen "Schlüssel" nach jedem Zeichen "rotiert". Rotor-basierte Verfahren sind längeninvariant. Unter einem DES Kryptifizierungsverfahren versteht man ein blockweises Verfahren, das ebenfalls auf Permutationen beruht.

Eine ähnliche Situation findet sich bei Komprimierungsverfahren, wo man grob zwischen verlustbehafteten (für Datenbanken unbrauchbaren) und verlustfreien Verfahren unterscheidet. Ein anderes Unterscheidungsmerkmal bil-

det die Geschwindigkeit der Komprimierung bzw. der Dekomprimierung. Gängige Verfahren sind: Lauflängenkomprimierung, LZW sowie Mischformen.

2. Authentifizierung

Jeder Datenbankzugriff erfolgt im Rahmen einer Connection, deren integraler Bestandteil eine Login-Authentifizierung ist. Seitens Transbase® werden keine Richtlinien für Passwörter forciert, d.h. der Benutzer muss sicherstellen, dass sein Passwort "sicher genug" ist.

Während der Authentifizierung wird das Paar (Username, Passwort) zum Server geschickt (im Klartext oder verschlüsselt, s.u.). Der Transbase® Server wendet nun einen DES-artigen Verschlüsselungsalgorithmus auf das Passwort an und vergleicht das verschlüsselte Passwort mit dem entsprechenden, zu Username gehörenden `sysuser`-Eintrag. Bei Nichtübereinstimmung wird der `CONNECT`-Wunsch verweigert. Die Tabelle `sysuser` speichert das Passwort also nicht im Klartext, sondern verschlüsselt. Bei eingeschalteter Seitenverschlüsselung wird es sogar zweimal verschlüsselt.

Der DES-Algorithmus hat einen festen, im Code enthaltenen 2-Byte-String als sogenanntes "Salt".

Da die Authentifizierung algorithmisch erfolgt, ist natürlich eine Umgehung mittels Codemanipulation möglich (sehr einfach durch Transaction bzw. Besitzer des Quellcodes). LOGIN Requests können protokolliert werden (Account Log).

3. Kommunikationsmedium

Der Austausch von Daten zwischen Client und Transbase® Server erfolgt über TCP/IP Sockets. Beim Kreieren der Datenbank (und auch zu einem späteren Zeitpunkt) kann per `tbadmin` die Verschlüsselung der Kommunikation ein- und ausgeschaltet werden.

Bei eingeschalteter Kommunikationsverschlüsselung wird auf beiden Seiten ein "Rotor"-basiertes Verfahren verwendet, welches auf beiden Seiten gleich initialisiert wird. Die Initialisierung erfolgt durch einen Vektor der Länge 16, der im Code (von `tbkernel.exe` und `tbx.dll`) hinterlegt ist.

4. Speichermedium

4.1. Festplatte

Beim Kreieren einer Datenbank kann festgelegt werden, dass die Speicherung (des Datenbankinhalts, aber auch von Logfiles) verschlüsselt erfolgt. Da der

Zugriff auf die Datenbank immer seitenweise erfolgt, erfolgt auch die Verschlüsselung seitenweise.

Der verwendete Algorithmus ist rotor-basiert. Die Initialisierung des Algorithmus erfolgt - ähnlich wie bei 3 - durch einen Vektor der Länge 16, der im Code (von `tbkernel.exe`) hinterlegt ist. Allerdings wird der Schlüssel noch mit der Seitennummer modifiziert, so dass jede Seite anders verschlüsselt wird.

4.2. RO-Medien

Wahlweise ist für RO-Medien zusätzlich eine Komprimierung möglich. Wenn Komprimierung und Verschlüsselung aktiv sind, so wird zunächst komprimiert und dann verschlüsselt (bzw. zunächst entschlüsselt und dann dekomprimiert). Dadurch erhält das Komprimierungsverfahren einerseits seine Leistungsfähigkeit, andererseits sind Angriffe auf das verschlüsselte Ergebnis mittels stochastischer Verfahren praktisch erfolglos.

Die Komprimierung kann beim Erstellen der ROM-Files spezifiziert werden. Die Verschlüsselung erfolgt, wenn die Editorial-Datenbank verschlüsselt war.

Das Komprimierungsverfahren ist ein proprietäres Verfahren, welches Elemente aus ZIP und LZW kombiniert. Das Verfahren wurde für die (häufig benötigte) Dekomprimierung optimiert, während die (nur beim Erstellen von ROM-Files benötigte) Komprimierung relativ langsam ist.

5. Mögliche Erweiterungen

Um Transbase® Datenbankinhalte noch besser gegen Angriffe zu schützen, wären folgende Erweiterungen denkbar:

1. Die Initialisierung der Algorithmen aus 3 und 4.1 könnte von aussen erfolgen, so dass sie sich nicht im Code findet. Diese Initialisierung könnte dann z.B. bei jedem Bootvorgang oder bei jedem Connect von aussen abgefragt werden. Hier wäre auch denkbar, dass dieser Initialisierungsparameter von externen Datenträgern, wie z.B. einer Smartcard, gelesen wird. Ohne diesen Initialisierungsparameter wäre die Datenbank nicht einmal durch Transaction zu lesen.

2. Der Transbase® Server könnte ein Public-Key Verfahren verwenden, um gleichlaufende Initialisierungen zwischen Client und Server zu erreichen: Der Client könnte einen Initialisierungsstring erzeugen, mit dem Transbase® PublicKey verschlüsseln und zum Server übertragen, wo er entschlüsselt werden kann. Mit diesem so übermittelten Initialisierungsstring wird dann die Session verschlüsselt.

3. Nach einem fehlerhaften CONNECT könnte leicht eine künstliche Verzögerung eingebaut werden.

6. Zertifizierung

Die benutzten Verfahren sind nicht zertifiziert im Sinne einer Überprüfung durch Dritte (z.B. BSI). Der Einsatz von zertifizierten, öffentlich zugänglichen, lizenzfreien Verschlüsselungsverfahren wäre relativ einfach möglich, solange sie symmetrisch, längeninvariant und plattformunabhängig arbeiten.

Kontakt

Transaction Software GmbH
Willy-Brandt-Allee 2
81829 München

Tel.: +49 89 / 627 09 - 0
Fax: +49 89 / 627 09 - 11

info@transaction.de
www.transaction.de
www.transbase.de