credativ

Sicherer PostgreSQL-Betrieb

Nach BSI-Grundschutz

Michael Banck <michael.banck@credativ.de>

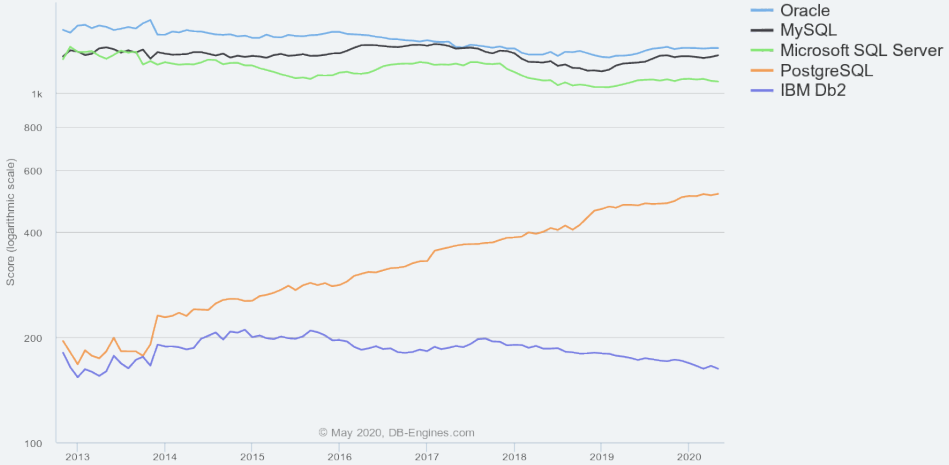
FrOSCon 2020 - Cloud Edition

PostgreSQL

credativ

- "*The World's Most Advanced Open Source Relational Database*"
- Hersteller-Unabhängig, kommerzieller Support von mehreren Firmen erhältlich
- BSD-Lizenz, Keine Copyright-Assignments, Open-Core oder Dual-Lizenzierung
- Focus auf Datenintegrität, kompromisslos stabil und zuverlässig
- Gute und konsistente Abdeckung des SQL-Standards, sinnvolle moderne Erweiterungen
- Bietet die meisten gängigen Enterprise-Features
- Viele Erweiterungen, prozedurale Sprachen und Treiber

DB-Engines Ranking of Relational DBMS



https://db-engines.com/en/ranking_trend/relational+dbms/all

BSI IT-Grundschutz

`https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz_node.html`

BSI IT-Grundschutz

credativ

> *”Als IT-Grundschutz bezeichnet die Bundesverwaltung eine vom Bundesamt für Sicherheit in der Informationstechnik (BSI) entwickelte Vorgehensweise zum Identifizieren und Umsetzen von Sicherheitsmaßnahmen der unternehmenseigenen Informationstechnik (IT).”*

`https://de.wikipedia.org/wiki/IT-Grundschutz`

Alternative internationale Konzepte

- PCI Compliance
- Common Criteria
- DISA STIG

UNCLASSIFIED

PostgreSQL 9.x Security Technical Implementation Guide

Version: 1

20 Jan 2017

BSI IT-Grundschutz

- Üblicherweise ein Konzern/Behörden-weites Projekt
  - Projekt-Team
  - Externe Unterstützung/Schulung
- Verschiedene (teils jährlich wieder kehrende) Phasen
  - Planungs-Phase, Schutzbedarfsfeststellung
  - Konzepterstellungs-Phase
  - Umsetzungs- und Dokumentations-Phase
  - Auditierung (von Teilen)
  - Zertifizierung (von Teilen)
- DBAs sind als Teil des Projekts in bestimmte relevante Bausteine eingebunden
- Evtl. Zusammenarbeit mit Kunden (insbesondere wenn Zertifizierung von diesen verlangt)

”Die Leitungsebene MUSS die Gesamtverantwortung für Informationssicherheit in der Institution übernehmen, und zwar so, dass dies für alle Beteiligten deutlich erkennbar ist. Die Leitungsebene der Institution MUSS den Sicherheitsprozess initiieren, steuern und kontrollieren. Die Leitungsebene MUSS Informationssicherheit vorleben.”

”Die Behörden- bzw. Unternehmensleitung MUSS die Zuständigkeiten für Informationssicherheit festlegen. Die zuständigen Mitarbeiter MÜSSEN mit den erforderlichen Kompetenzen und Ressourcen ausgestattet werden.”

BSI IT-Grundschutz

- Elementare Gefährdungen
- Bausteine
- Gefährdungslage
- Anforderungen (Maßnahmen)
  - Basis-Anforderungen (MUSS)
    - Unbedingt erforderlich
  - Standard-Anforderungen (SOLLTE)
    - Normalerweise umzusetzen
    - Gründe können dagegen sprechen
    - Sorgfältige Abwägung und Begründung
  - Anforderungen bei erhöhtem Schutzbedarf
- Umsetzungshinweise

Bausteine

IT-Grundschutz in verschiedene Bausteine aufgeteilt

- ISMS: Sicherheitsmanagement
- ORP: Organisation und Personal
- CON: Konzeption und Vorgehensweise
- OPS: Betrieb
- DER: Detektion und Reaktion
- APP: Anwendungen
- SYS: IT-Systeme
- IND: Industrielle IT
- NET: Netze und Kommunikation
- INF: Infrastruktur

Bausteine

Potenziell relevante Unter-Bausteine:

- OPS.1.1.2 Ordnungsgemäße IT-Administration
- OPS.1.1.3 Patch- und Änderungsmanagement
- OPS.1.1.4 Schutz vor Schadprogrammen
- OPS.1.1.5 Protokollierung
- OPS.2.2 Cloud-Nutzung
- DER.3.1 Audits und Revisionen
- DER.4 Notfallmanagement
- APP.4.3 Relationale Datenbanksysteme
- SYS.1.1 Allgemeiner Server
- SYS.1.3 Server unter Linux und Unix
- SYS.1.5 Virtualisierung

Bausteine

Potenziell relevante Unter-Bausteine:

- OPS.1.1.2 Ordnungsgemäße IT-Administration
- OPS.1.1.3 Patch- und Änderungsmanagement
- OPS.1.1.4 Schutz vor Schadprogrammen
- OPS.1.1.5 Protokollierung
- OPS.2.2 Cloud-Nutzung
- DER.3.1 Audits und Revisionen
- DER.4 Notfallmanagement
- **APP.4.3 Relationale Datenbanksysteme**
- SYS.1.1 Allgemeiner Server
- SYS.1.3 Server unter Linux und Unix
- SYS.1.5 Virtualisierung

Baustein APP.4.3 “Relationale Datenbanksysteme”

- Gefährdungslage
  - 2.1 Unzureichende Dimensionierung der Systemressourcen
  - 2.2 Aktivierte Standard-Benutzerkonten
  - 2.3 Unzureichende Vergabe von Berechtigungen
  - 2.4 Unverschlüsselte Datenbankanbindung
  - 2.5 Datenverlust in der Datenbank
  - 2 6 Integritätsverlust der gespeicherten Daten
  - 2.7 SQL-Injections
  - 2.8 Unzureichendes Patchmanagement
  - 2.9 Unsichere Konfiguration des Datenbankmanagementsystems
  - 2.10 Malware und unsichere Datenbank-Skripte

Schutzbedarf

- Verletzung der Grundwerte
  - Vertraulichkeit
  - Integrität
  - Verfügbarkeit
- Auswirkung auf Aufgabenerfüllung und Geschäftstätigkeit bei Vorfällen/Risiken
- Drei Stufen der Schadensauswirkung
  - Normal - begrenzt und überschaubar
  - Hoch - möglicherweise beträchtlich
  - Sehr Hoch - möglicherweise existentiell bedrohlich, katastrophal
- Im Allgemeinen normaler Schutzbedarf
- Schutzbedarfsfeststellung

Generelle Betrachtungen

Planung und Umsetzung

- Modellierung des Schutzbedarfs aller Komponenten, Maximalprinzip
- Entwickeln konkreter Sicherheitsmaßnahmen
  - Dokumentation, warum Maßnahmen ausreichend sind
  - Sortierung der Maßnahmen nach Priorität
  - Umsetzungsplan aufführen
  - Verantwortliche definieren - wer ist verantwortlich für deren
    - Initialisierung
    - Umsetzung
    - Kontrolle (z.B. Audit)
    - Revision
- Umsetzung zeitnah garantiert? Kontrolliert und protokolliert?
- Angemessenheits- und Wirtschaftlichkeits-Betrachtung
  - Wie teuer sind welche Maßnahmen?
  - Ist das sinnvoll? Wer hat die Entscheidung notfalls zu verantworten?

Benötigte Konzepte

credativ

Nötig

- Sicherheitskonzept
- Benutzer- und Berechtigungskonzept
- Datensicherungskonzept
- Protokollierungskonzept

Sinnvoll

- Logauswertungskonzept
- Monitoringkonzept

Trennung der Arbeitsbereiche TDBA/FDBA

- Technische DBAs (TDBAs):
  - Einrichtung Server, Instanzen, Backups
  - Erstellung Nutzer/Gruppen und Datenbanken
  - Evtl. Erstellung Schemas
- Fachliche DBAs (FDBAs):
  - Erstellung Datenbank-Objekte
  - Verwendung FDBA/Schema-Owner Rolle (`SET ROLE`)
  - Tuning von Abfragen
- Auftragsberechtigter
  - Klare Benennung, wer von fachlicher Seite aus Änderungen beantragen darf
- In der Praxis gibt es oft keine FDBAs und die Auftragsberechtigten sind Liaison zu externen Entwicklern/Software-Hersteller
- Systemadministration
  - Root-Kennung kann von anderer Gruppe (Systemverwaltung) betreut werden
  - Start/Stopp von PostgreSQL per sudo für TDBAs

Separierung von Test- und Produktivdaten

credativ

- Mögliche Separierung in PostgreSQL:
  - Host/VM/Container
  - Instanz
  - Datenbank
    - Rollen sichtbar
  - Schema
    - Objekte sichtbar
- Produktive Instanzen sollten auf eigenem Server laufen
- Entwicklungs/Test-Instanzen sollten auf eigenem Server laufen
- Verschiedene Kunden sollten (wenn möglich) jeweils eigene Instanzen erhalten
- Ressourcen-Management ohne VMs/Container bei vielen Instanzen schwierig

Notfall-Nutzer/Jump-Host/Plan/Übung

credativ

- Notfall-Nutzer
  - Für Bereitschafts-Mitarbeiter, die keinen regulären Zugang haben
  - Passwort in Keystore oder Tresor gelagert
- Notfall-Jump-Host
  - Alternativer Jump-Host, falls üblicher Jump-Host Down ist
- Notfall-Plan
  - Anweisungen für den Notfall
  - Alarmierungs/Eskalationskette
- Notfall-Übung
  - Typischerweise einmal im Jahr
  - Test des Notfall-Plans

Datenbank Lebenszyklus

credativ

”Neue Datenbanken MÜSSEN nach einem definierten Prozess angelegt werden. Wenn eine neue Datenbank angelegt wird, MÜSSEN Grundinformationen zur Datenbank nachvollziehbar dokumentiert werden.”

”Es MUSS ein Prozess etabliert werden, der regelt, wie Datenbankbenutzer und deren Berechtigungen angelegt, genehmigt, eingerichtet, modifiziert und wieder entzogen bzw. gelöscht werden”

Datenbank Lebenszyklus

- Beantragung
  - Per Formular, Ticket o.ä.
    - Möglichst wenig Auswahl
- Inbetriebnahme
  - Check-Liste, Playbook etc.
  - Abnahme durch Kunden
- Änderungen
  - Neue Datenbanken, Nutzer
  - Zusätzliche Erweiterungen
  - Konfigurations-Änderungen
- Patching, Major Upgrade
- Aussonderung
  - Check-Liste, Playbook etc.

Betriebshandbuch

credativ

- Beschreibung der Arbeitsabläufe
- Am Besten keine dynamischen Daten (aktuelle Liste der Datenbanken o.ä.)
- Bestimmte Bereiche (z.B. Monitoring) können ausgelagert sein
- Checklisten für übliche Arbeitsabläufe
  - Installation/Entfernung von Instanzen
  - Falls kein Konfigurations-Management vorhanden

Dokumentation von Änderungen/Störungen

Relativ egal wie

- Sharepoint
- Ausgefüllte PDF-Formulare
- Text-Dateien
- Tickets
- Git

Verwendete Version und Patch-Plan

”Es MUSS sichergestellt sein, dass die Installationspakete des Datenbankmanagementsystems aus sicheren Quellen stammen. Bereits veröffentlichte Patches MÜSSEN eingespielt werden, bevor das DBMS betrieben wird.”

”Vorhandene Sicherheitsupdates für das Datenbankmanagementsystem [...] MÜSSEN zeitnah installiert werden. Vorab MUSS auf einem Testsystem überprüft werden, ob die Sicherheitsupdates kompatibel sind und keine Fehler verursachen.”

”Des Weiteren MUSS geprüft werden, ob die Update-Intervalle des Datenbankmanagementsystems auf die Update-Zyklen des Herstellers abgestimmt werden können.”

Verwendete Version und Patch-Plan

- PostgreSQL Major-Releases werden 5 Jahre lang unterstützt
  - Neue Major-Versionen nach 2-3 Point Releases bereit für Produktiv-Einsatz
  - Vierteljährliche, planbare Point-Releases für kritische/sicherheitsrelevante Bugs
  - Jeweils am zweiten Donnerstag des zweiten Monats eines Quartals
  - Zeitplan: `https://www.postgresql.org/developer/roadmap/`
  - Security Team für Sicherheitsvorfälle; Notfalls außerordentliche Point-Releases
- Distributions-Pakete für alle supporteten Versionen für Red Hat / CentOS / SLES und Debian / Ubuntu
  - `http://yum.postgresql.org`
  - `http://apt.postgresql.org`
- Patchen benötigt Einspielen der Pakete und Neustart der Instanz
  - Beendet laufende Abfragen und Sessions
  - Wartungsfenster oder organisatorischer Vorlauf (inkl. Ansprechpartner) nötig
  - Zunächst Test-Instanzen, z.B. eine Woche später Produktions-Instanzen

Härtung der Standard-Berechtigungen

”Das Datenbankmanagementsystem MUSS gehärtet werden. Hierfür MUSS eine Checkliste mit den durchzuführenden Schritten zusammengestellt und abgearbeitet werden.”

”Die Basishärtung MUSS regelmäßig überprüft und, falls erforderlich, angepasst werden.”

Härtung der Standard-Berechtigungen

- PostgreSQL erlaubt standardmäßig allen Nutzern sich an Datenbanken anzumelden
  - `REVOKE ALL ON DATABASE dbname FROM PUBLIC;`
  - `GRANT CONNECT ON DATABASE dbname TO "user-Group";`
- PostgreSQL erlaubt standardmäßig allen Nutzern im `public`-Schema Objekte anzulegen
  - `REVOKE ALL ON SCHEMA public FROM PUBLIC;`
- Manchmal können Standard-Anwendungen können nur mit dem `public`-Schema umgehen, Ausnahmen nötig

credativ

System- und Datenbank-Zugriff

System-Zugriff

credativ

- Der `postgres`-Nutzer sollte sich nicht via SSH einloggen dürfen
- TDBA muss sich mit persönlichem Account von Jump-Host aus einloggen
  - Wenn nötig dann Aufschaltung auf `postres`-Nutzer via sudo
- Jump-Host sollte nur via VPN erreichbar sein, nicht direkt per SSH
- Keinen direkten Root-Login erlauben
  - In der Praxis wird Root/Admin-Account oft von anderen Teams verwaltet

postgres-Systembenutzer

- `postgres`-Systembenutzer (oder Instanz-Besitzer) wird für Starten/Stoppen der Instanz benötigt
- PostgreSQL Daten-Verzeichnis kann nur von `postgres`-Nutzer gelesen werden
  - Ab v11 kann einer Gruppe Zugriff gewährt werden
- Am Besten Einschränkung auf konkrete Sudo-Befehle, keine allgemeine Shell
  - In der Praxis oft schlecht umsetzbar

Kennungen und Passwörter

”Alle Passwörter der Datenbankbenutzer MÜSSEN der Passwortrichtlinie der Institution entsprechen.”

”Das Benutzer- und Berechtigungskonzept der Institution MUSS um die für Datenbankmanagementsysteme notwendigen Berechtigungen für Rollen, Profile und Benutzergruppen erweitert werden.”

”Es MUSS ein Prozess etabliert werden, der regelt, wie Datenbankbenutzer und deren Berechtigungen angelegt, genehmigt, eingerichtet, modifiziert und wieder entzogen bzw. gelöscht werden.”

Kennungen und Passwörter

- Passwort-Manager für generelle Passwörter auf TDBA-Seite
- Postgres bietet keine Möglichkeiten für Passwort-Richtlinien wie Komplexität, oder regelmäßige Änderungen
- Technische Nutzer für Anwendungsserver usw.
  - SCRAM Passwörter/Authentifizierung (`scram-sha-256`)
  - Regelmäßige Passwort-Wechsel technisch schwer zu überprüfen
  - FDBAs/Anwendungsbetreuer sollten Kenntnis/Absicht darüber bestätigen
- Personalisierte Nutzer
  - Einer bestimmten Person zugeordnet
  - Keine Gruppen-Accounts
  - Passwort-Richtlinien via LDAP/Active-Directory (`ldap`) umsetzbar
    - Verschlüsselte Verbindung, da LDAP-Passwort im Klartext vom Client übertragen wird
  - Alternativ `gssapi` für komplett verschlüsselte Authentifizierung mit Active Directory
    - Kompliziert (Kerberos), vor allem wenn keine Kontrolle über AD-Server besteht

IP-Freischaltungen und Verbindungsaufbau

credativ

”Das Datenbankmanagementsystem SOLLTE so konfiguriert werden, dass Datenbankverbindungen immer verschlüsselt werden.”

IP-Freischaltungen und Verbindungsaufbau

- Zugriff für TDBAs nur über personalisierte Kennung oder `local` via `peer`
- Nur benötigte Quell-IP-Adressen in `pg_hba.conf` freischalten
  - Bzw. IP-Bereiche nach Minimal-Prinzip
- TCP/IP Datenbank-Verbindungen sollten generell verschlüsselt sein (`hostssl`)
  - Seit v10 ist ein Austausch des Server-Zertifikats ohne Neustart der Instanz möglich
- Feingranulare Einstellungen von Rollen und Datenbanken möglich, aber in der Praxis oft schnell unübersichtlich
  - Unterscheidung IP-Adressen von technischen Nutzern (`scram-sha-256`) und personalisierten Kennungen (z.B. `ldap`)
  - Vergabe von Gruppen an die entsprechenden Rollen z.B. `scram-Group` und `ldap-Group` und Verwendung in `pg_hba.conf`

Beispiel pg_hba.conf

```
# TYPE  DATABASE USER         ADDRESS         METHOD
local   all      postgres                     peer
local   all      +tdba-Group                  peer

hostssl all      +scram-Group 10.1.0.23/32    scram-sha-256
hostssl all      +scram-Group 10.1.0.65/30    scram-sha-256

hostssl all      +ldap-Group  172.16.43.75/24 ldap
    ldapserver=domain.foo.de ldapprefix="FOO\" ldapport="389"
    ldapsuffix="" ldaptls=1
hostssl all      +ldap-Group  172.16.13.88/24 ldap
    ldapserver=domain.foo.de ldapprefix="FOO\" ldapport="389"
    ldapsuffix="" ldaptls=1
```

credativ

Sicherer Betrieb

Protokollierung

credativ

”Sicherheitsrelevante Ereignisse des Datenbanksystems MÜSSEN mit einem eindeutigen Zeitstempel protokolliert werden. Dabei MÜSSEN sich Art und Umfang der Protokollierung am Schutzbedarf der zu verarbeitenden Informationen orientieren.”

”Es SOLLTE so protokolliert werden, dass die Protokolldateien nicht nachträglich verändert werden können.”

Protokollierung

- Protokollierungs-Konzept
- Revisionssichere Logdateien
- Versand via syslog-ng an Log-Server (verschlüsselt)
  - `log_destination = 'stderr,syslog'`
  - `syslog_facility = 'local7'`
  - `syslog_ident = 'postgres-$HOST-$INSTANZ'`
- TDBAs haben dort höchstens Lese-Rechte
- Möglichkeit FDBAs die Logdateien (ohne SSH-Zugang zum DB-Server) bereitzustellen
- Aufbewahrungsfristen beachten (30 Tage üblich)

Auditierung

credativ

- Logging von Verbindungen (`log_connections`/`log_disconnections`)
  - Leider unflexibel; oft sehr viel Rauschen durch technische Nutzer
- Logging von DDL / Konfigurations-Änderungen (`log_statement = 'ddl'`)
- Logging von DML der TDBAs (`log_statement = 'mod'` pro TDBA Rolle)
  - Eigenschutz
- Bei Instanzen mit hohem hohem Schutzbedarf `log_statement = 'mod'`
- Bei Instanzen mit hohem sehr hohem Schutzbedarf `log_statement = 'all'`
- Wenn möglich `pgaudit`-Erweiterung für gezieltere Auditierung verwenden
  - Audit-Ereignisse werden in das normale Postgres-Log geschrieben
- Auditierung von Konfigurations-Änderungen nicht einfach
  - Änderungen während einer Downtime werden nicht geloggt
  - Periodische Auditierung der `postgresql.conf` Werte gegen SOLL-Daten

Backup/Restore(-Tests)

credativ

”Es MÜSSEN regelmäßig Systemsicherungen des DBMS und der Daten durchgeführt werden. […] Hierfür SOLLTEN die dafür zulässigen Dienstprogramme benutzt werden.”

”Alle Transaktionen SOLLTEN so gesichert werden, dass sie jederzeit wiederherstellbar sind.”

”Die vorgenommenen Datensicherungen SOLLTEN regelmäßig daraufhin überprüft werden, ob die Integrität der Sicherungsdateien noch gewährleistet ist. Die verantwortlichen Mitarbeiter SOLLTEN zudem regelmäßig üben, wie sich Datenbanken im Notfall schnell wiederherstellen lassen.”

Backup/Restore(-Tests)

credativ

- Datensicherungs-Konzept
- Ab einer gewissen Größe physikalische Backups nötig
  - Sinnvoll ein fertiges Produkt wie pgBackRest, Barman zu verwenden
  - Bei kleineren Instanzen auch Dumps (oft lokale Skripte)
- Definierte Retention-Time
  - Üblich 2-5 Tage, Auslagerung auf externe Storage-Lösungen
  - Evtl. Aufbewahrung Monats/Jahressicherungen
  - Beachtung evtl. Archivierungs-Pflichten des letzten Backups nach Aussonderung
- Automatisierte Restore-Tests der produktiven Instanzen (z.B. jede Woche)
  - Bei Point-in-Time-Recovery nachfolgender Dump empfohlen

Monitoring

credativ

”Es SOLLTEN Parameter, Ereignisse und Betriebszustände des Datenbankmanagementsystems definiert werden, die für den sicheren Betrieb kritisch sind. Diese SOLLTEN mithilfe eines Monitoring-Systems überwacht werden. Für alle kritischen Parameter und Ereignisse SOLLTEN Schwellwerte festgelegt werden. Wenn diese Werte überschritten werden, MUSS geeignet reagiert werden.”

Monitoring

- Oftmals zentrales Monitoring vorhanden
  - Manchmal für TDBAs nicht verfügbar
  - Oder irgendein Enterprise-Tool ohne Postgres-Integration
- Icinga-Monitoring mit Alerting im Prinzip ausreichend
- Zusätzliche Checks/Metriken hilfreich
  - `check_postgres`
  - InfluxDB/Grafana für Graphen
- Alternativ
  - Prometheus
  - `{node,sql,postgres}_exporter`
  - AlertManager
  - Grafana Dashboards

Schlussbemerkungen

- Ab einer gewissen Größe sind viele der Anforderungen sinnvoll
- Selbstschutz für DBAs
  - Auftragsberechtigte
  - Paper Trail
  - Protokollierung/Auditierung
- Nicht die BSI-Keule schwingen
- Umsetzung mit Augenmaß
  - Vieles lässt sich (sinnvoll) Verargumentieren oder Begründen
- Argumentations-Grundlage gegenüber Kunden
  - “Das muss Ihr ISB freigben / Ihr Behördenleiter abzeichen”
  - “Da brauchen wir eine Risiko-Übernahme”

Risiko-Übernahme

https://twitter.com/derpupe/status/963496089608441856

Danke für die Aufmerksamkeit

- Fragen?
- Michael Banck <michael.banck@credativ.de>
- https://www.credativ.de
- https://www.credativ.de/postgresql-competence-center
- https://www.credativ.de/karriere
- https://www.credativ.de/blog
- https://github.com/credativ