Oracle Forensics

An Introduction into Oracle Forensics, Tamper Detection and Log Analysis

Dr. Günter Unbescheid

Database Consult GmbH, Jachenau

Database Consult GmbH

- Founded in 1996
- Specialized in ORACLE-based Systems
- Focus Areas
  - Security, Identity Management
  - Tuning, Installation, Configuration, Systemanalysis
  - Support, Troubleshooting, DBA-Tasks
  - Databasedesign, Datamodelling und –design
  - Custom made Workshops
  - www.database-consult.de
- Since 2012 – Collaboration with trivadis makes IT easier.

„**Es gibt nicht einen 'besten' Weg** oder Universal-Ansatz für jedes Problem. Die eigentliche Herausforderung in der IT ist nicht das Erlernen des besten Weges, sondern das Erlernen so vieler verschiedener Wege wie nur möglich, um beurteilen zu können, welcher Weg unter den gegebenen Umständen der beste ist."

Tom Kyte's bester Rat nach 30 Jahren IT-Karriere (DOAG Online von 22.9.2015)

We define **computer forensics** as the discipline that combines elements of law and computer science to collect and analyze data from computer systems, networks, wireless communications, and storage devices in a way that is admissible as evidence in a court of law.
… main goal of computer forensics is to identify, collect, preserve, and analyze data …
(US CERT)

Database forensics is a branch of digital forensic science relating to the forensic study of databases and their related metadata.
(Wiki)

The forensic analysis of a **compromised database server** presents its own unique challenges. In other areas of computer forensics it's often obvious that a crime has been committed: pornographic images are discovered on a hard drive; a rootkit has been installed; a system has been trashed.

In the case of a database intrusion however it may appear at first glance that nothing untoward has happened - *prima facie* evidence appears absent.

In the physical world if something is stolen it is gone and by that the theft becomes obvious but with computers, and specifically database servers, when data is stolen, only a copy is taken and the original remains.

David Litchfield

Objectives

- Introduction to basic forensic investigation methods
- Preparatory measures for database systems
- Datasources and –formats for database forensic analysis
- Realize possible security relevant incidents in due time
- Keep in mind
  - Database Forensics is only a part of a comprehensive IT forensic analysis, which is not covered in this lecture
  - Has to be carried out with external expertise

Agenda

- Forensic Fundamentals
- Types of Breaches
- Types of Tools
- System/Database Preparation
- Auditing and Logserver Configuration
- Timelines and Timformats
- Database Forensic Sources
- Most Important Database Objects

Forensic Fundamentals

- Analysis is complex and has to combine many different sources – both at OS- and DB-levels
  - Can be triggered by criminal activities or normal incident-response
- Data and findings have to be stored in revision-proof manner
  - Use (physically) separate analysis systems
  - Store data read only for subsequent securing of evidence
  - Wirte and preserve a detailed protocoll of actions
- Since (mostly) personal data is involved data privacy laws have to be followed and data protection legislation will apply
- Scope
  - Offline or postmortem analysis collects non-volantile data
  - Live or online analysis collects volatile data, e.g. process lists, memory content, network connections

Forensic Fundamentals

- Data collection should be minimal invasive in order to preserve volatile data to the maximum extent possible and avoid accidentally destroying data
- Basic questions:
  - What happened, where, when and how
  - Who has done it, can it be repeated, what will be the damage
- Possible strategies have to be carefully considered and balanced
  - Take system offline – lose volantile data and throw away chances to analyse the incident, undermine availability, stop suspicious actions for the moment
  - Keep System online – keep volatile data, retain availability, maybe encourage further tampering of data

Types of Breaches

- Exploitation by trust / Insider Threat – difficult to track
  - Using well known credentials
- Configuration Weakness
  - Default usernames and passwords, unlocked users
  - Unnecessary Services, e.g. external procedures, XDB etc.
- Software Vulnerability – need for regular patching
  - Buffer overflows – overwrite program control information
  - Format String Vulnerability – caused by unchecked user input
  - PL/SQL Injection – exploiting dynamic SQL, e.g. by creating functions
  - Trigger abuse – injection of code into triggers
  - Cursor snarfing – access to other peoples cursors which had not been closed properly (fixed)
  - External procedures – EXTPROC loads libraries and executes; 12c run as designated OS credential

After the Breach

- Data Exfiltration – get data „out“
  - Physical access to the system
  - Network access
    - In-band – using the same channel as getting in (easy to track)
    - Out-of-band – using different channels, e.g. HTTP
- Maintaining Access – implement “rootkits”
  - Changing definition of views
  - Modify code of existing PL/SQL objects, such as triggers
  - Grant high privileges
  - Alter permissions on privileged/dangerous code

SIEM and IDS

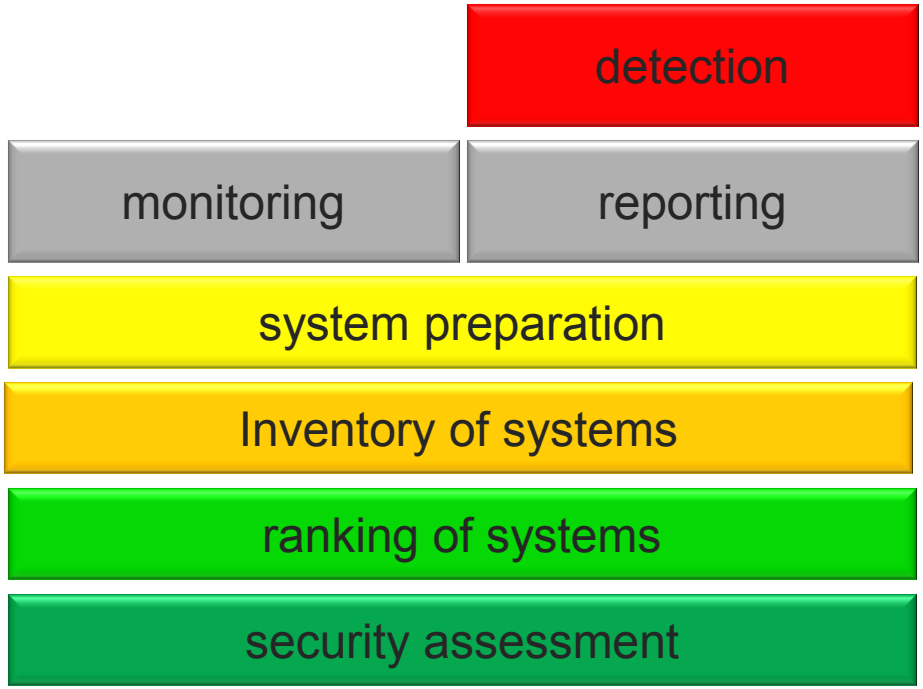
- SIEM – Security Information and Event Management
  - Hybrid solution handling security information (consolidation of logs, analysis of data) and security event management (managing threads, correlating data)
- IDS – Intrusion Detection System
  - device or software application that monitors network or system activities for malicious activities (host-based or network-based)
  - signature-based or statistical anomaly-based
- General Shortcomings
  - number of real attacks mostly far below the number of false-alarms
  - constantly changing library of signatures is needed to mitigate threats
  - cannot compensate for a weak identification and authentication mechanisms
  - Cannot analyze encrypted packages

SIEM and IDS

- Basically help to cut down time between atcks occur and incident response teams respond
- Not an all round carefree package
  - Mostly costing more than anticipated
  - Expertise often must be outsourced
  - Complex and time consuming setup before yielding results

In a survey of 800 IT professionals working for companies of all sizes involved in more than 30 industries, 74% of IT professionals who have deployed a SIEM (security information and event management) solution admitted that it didn't significantly reduced security incidents.
http://www.netwrix.com/SIEM_survey_2014.html

Preparatory „Stack“



- Proper preparations eases forensic analysis
- Assessment and ranking to reduce configuration effort
- Inventory for usage details
- System preparation to assure necessary data are available
- Monitoring to preserve the configuration
- Reporting for ad hoc analysis, can lead to
- Detection for systematic analysis and provisioning

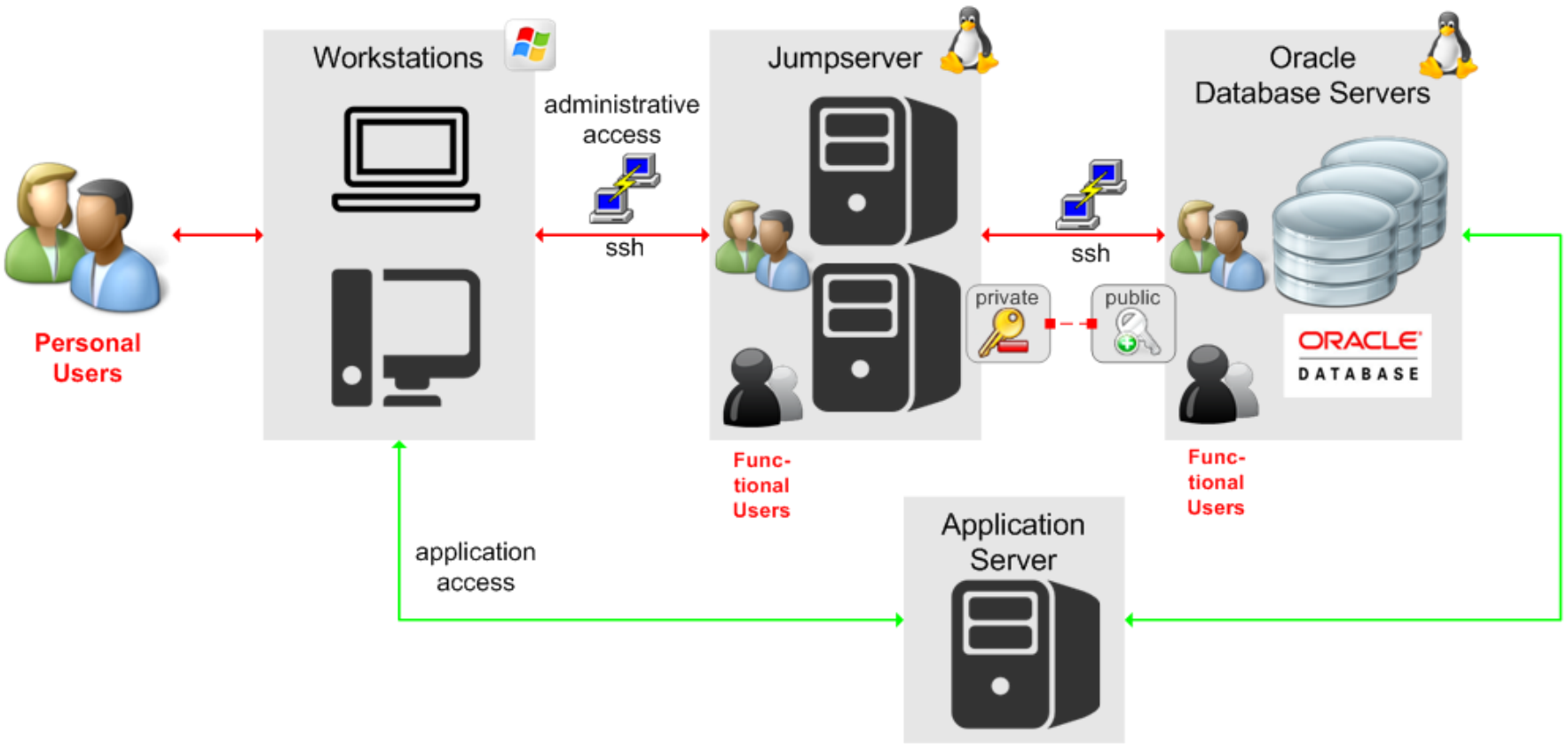
Preparing the System – OS

- Introduce a security ranking of systems
  - Reducing the cost and volume of protected systems
- Minimize the attack surface
  - Hardening: install only needed packages/services
- Implement proper OS Authentication mechanisms for administrators
  - Configure personal administrative accounts (endusers are usually accessing databases as remote clients)
  - For example where required SSH password protected keypair access from jump servers for connect efficiency
- Implement revision-proof and balanced OS-auditing
  - balancing data volume against security requirements
- Measures have to retain efficient corporate workflows

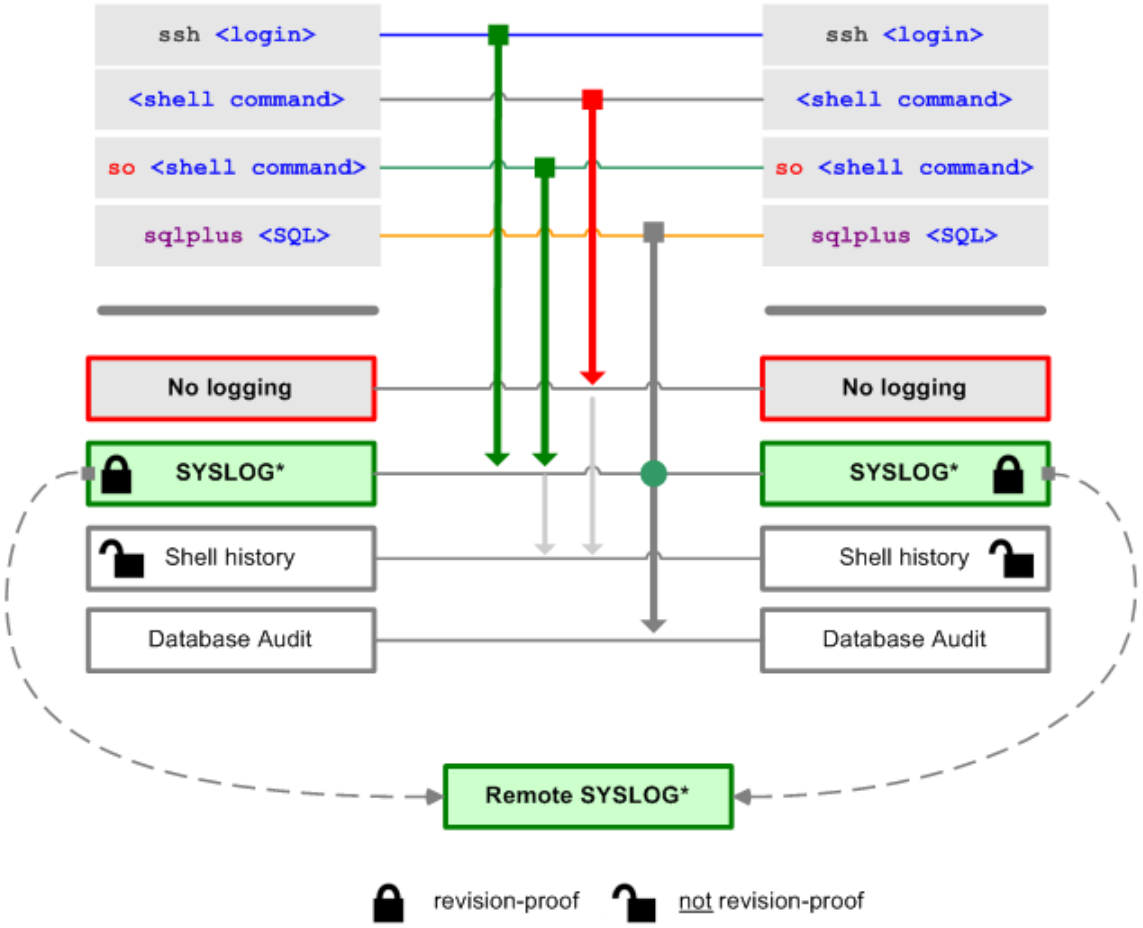
Preparing the System – OS

- Proper authorization of Oracle administrative commands at OS level
  - Reduce actions under Oracle login shell
  - E.g. authorization of privileged commands via sudo
- SUDO and SSH actions are logged by default
  - Syslog-ng or rsyslog are revision-proof for Oracle Admins
  - Operating a central SYSLOG server even enhances security and segregation of duties
- Linux offers several logging solutions – design with care!
  - PAM TTY kernel module – via AUDITD locally/SYSLOG remotely
  - TTY kernel logging as key logger – logs input including passwords
  - Psacct/acct log input locally, no SYSLOG plugin
  - Standard Shell logging/HISTFILE – data store only locally
  - ROOTSH or SUDOSH (shell wrapper) log input and output

Typical Oracle Landscape



Command Types and OS Logging



Preparing the System – Database

- Classify your data (and the databases storing them)
- Know your System
  - Applications/Servers/User groups accessing data
  - Most sensitive tables procedures, basic access patterns, coding standard
  - External procedures/tables, directory objects
- Implement proper authentication, (password policies, def. Accounts)
- Implement proper privilege profiles (need to know/least privilege)
- Care for traceability of actions (personal OS- and/or DB-Accounts)
- Configure revision proof auditing, preferrable at „central“ location
- Follow proper patching policies
- Consider encryption (network and for data-at-rest and bakups)

Preparing the System – Database

- Monitor your configuration
- Adminitrator access
  - Avoid external password file
  - Local connect AS SYSDBA via personal OS user
  - Personal administrative database user not always necessary
- Enduser Access
  - Personal DB-User or comparable measures, e.g. client identifier set by application server
- Prepare for Logminer usage
  - Configure Supplemental Logging

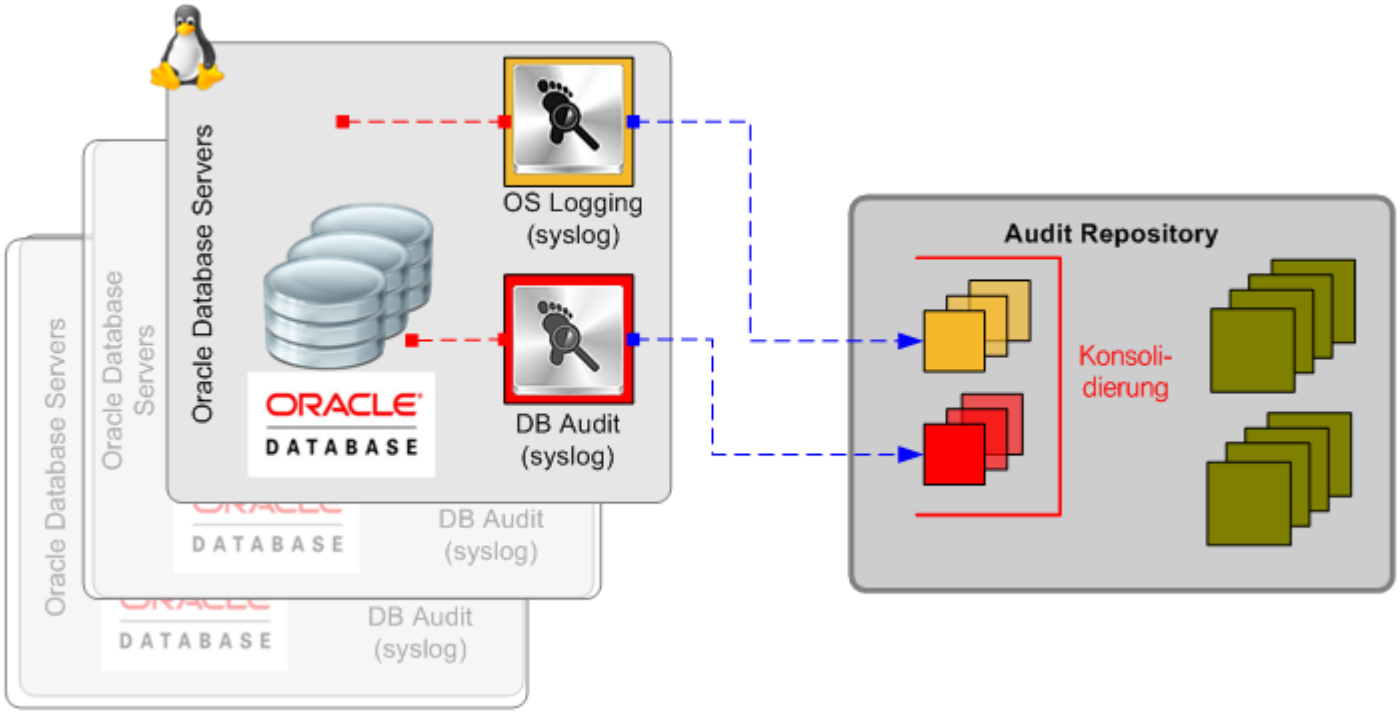
Database Auditing

- Auditing should be configured revision-proof and with evaluation in mind
- Classical auditing
  - SYS-auditing (`audit_sys_operations = true`) tracks complete commands
  - `audit all` + extra options for personal administrators
  - `audit_syslog_level` utilizes SYSLOG-NG for better segregation of duties and remote logging
  - BUG: `audit system audit` not working at all!
- Unified Auditing possible in 12c
  - Utilizing in-DB read only table, no SYSLOG support
  - audit policies for option bundeling
- Oracle Audit Vault and Database Firewall – software appliance (OEL)

Database Auditing

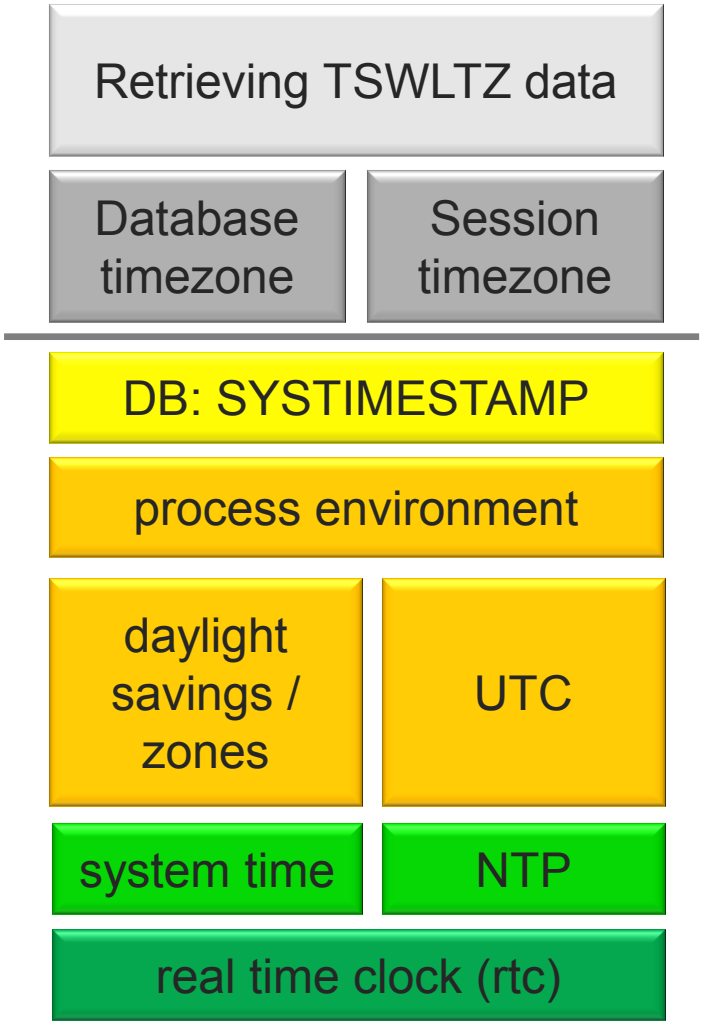
- Audit measures have to be negotiated with works council
- Recommendation: use SYSLOG-NG/RSYSLOG
  - Use particular <facility>/<level> for DB-Actions
  - Log into separate OS-file(s) and file system (space requirements!)
  - Use additional owner/group filters for other Oracle related OS admin actions
- Configure „central“ Log-Server(s) for Oracle-related files
  - Enhanced segregation of duties and audit safety
  - Position in separate network segment
  - Use TCP, encryption and compression for data transport
  - Use Hashes/checksums at logserver site for additional tamper proofing
- Different storage options at central site(s)
  - Raw and „parsed“ files, directory structures or DWH-usage

Central Logserver



Timelines, Timezones, Timeformats

- Forensic analysis is based on different log sources from different systems
  - Different formats/timezones
- Timelines arrange events from different sources chronologically
  - Depend on time synchronization
  - Optimize analysis process
- Hardware clock versus System Time
- DB Time depends on environment settings at startup
- Possible standardisation at logserver site when „parsed“



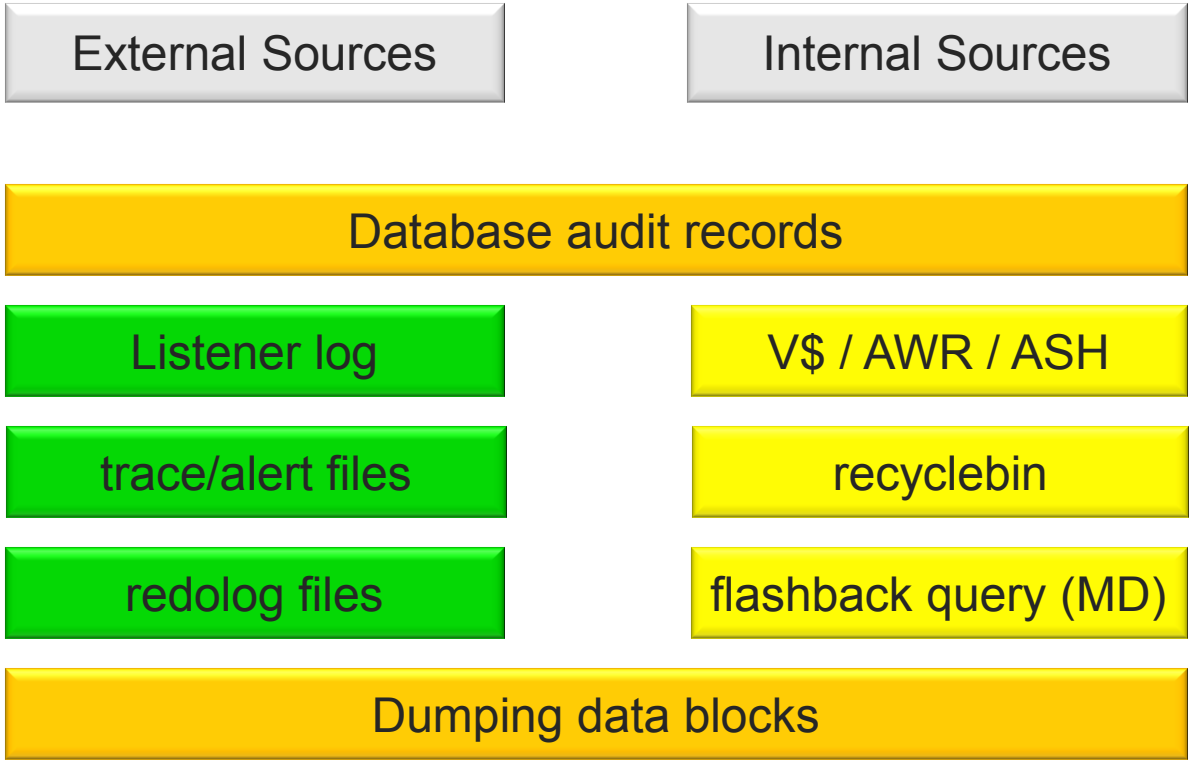
Timelines

- Synchronising time is a must in corporate networks
- Low Level
  - Use Network Time Procolls (NTP)
  - Signals can be distorted
- Medium Level
  - Configure addionally time signal receivers, e.g. DCF77
  - Receive official time signals
  - Signalscan be distorted
- High Level
  - Us a combination of time signal receiver and GPS receiver
  - Issue warnings when signals differ

Strategies in Case of (serious) Incidents

- Live response – recovering and safely storing volatile data for later analysis
  - As early as possible after suspicious event(s) are spotted.
  - Should be fully documented
  - Act minimally invasive and store results on remote filesystems
  - Obtain/record: system time, logged on users, users and groups, running processes, active ports, routes etc.
- After that collect database related sources
  - Volatile data – v$-Objects and the like
  - Persistent Data – external and internal sources, use base tables instead of views for internal data
- Be aware that attacker might fake tools, acts repeatedly
- Merge as many sources as possible (network, FW etc. not covered)

Database Forensic Sources



For forensic analysis we have to look into the past
There could be many traces of past actions
Some of them might be gone (volatile)

Comparing/Storing/Analyzing Datasets

- Result Sets of key data should be checked/monitored to detect tampering
  - PL/SQL objects: Trigger, Packages, Procedures etc
  - Roles and their privileges
  - Directory objects and external procedures/tables
  - Database Links
- Monitor by using Hash values, e.g.
  - SYS.DBMS_UTILITY.GET_HASH_VALUE
  - SYS. DBMS_SQLHASH.GETHASH
- Build Checksums of Result files
- Preserve entire table data in external tables using subqueries
  - Dump files can be transported to forensic system

Basic DB Objects for collection (samples)

- gvsql, gvdb_object_cache, gvsql_bind_data, gvsql_bind_capture
- col_usage$, mon_mod$
- aud$, fga_log$,
- user$
- WRH$_ACTIVE_SESSION_HISTORY , WRH$_SQLSTAT
- SYS.RECYCLEBIN$
- SYS.EXTERNAL_TAB$

Use these and related tables to build a baseline

Bottom Line …

- This lecture gave only a brief introduction into the topic
- Forensics in general and database forensics in particular are very complex
- In depth knowlege of the techniques is necessary for thorough analysis
- However, preparing the environment is „easy“ and
  - helps in detecting possible threats earlier
  - speeds up data analysis when needed
  - Supports day to day incident responses
- Building a proper environment involves internal and external resources

Thank you for listening

www.database-consult.de