



## Client Oriented Classification of DB Attacks and Countermeasures

Amichai Shulman, CTO, Imperva Inc.

# Introduction

- Database security today – A brief overview
- Evident pitfalls of current database security practices
- Database security threats revisited – Client Based Classification Scheme
- Client Based Attack Classification in details
- Matching the right solution to the relevant problem – Building effective and efficient countermeasures

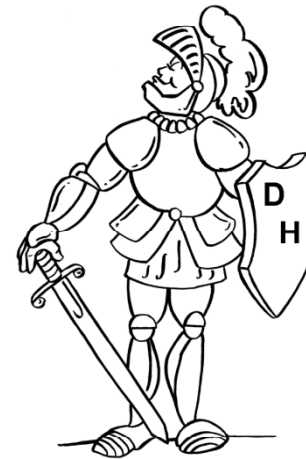
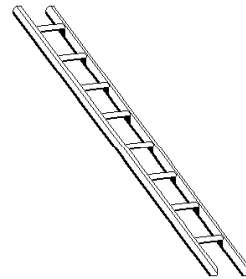
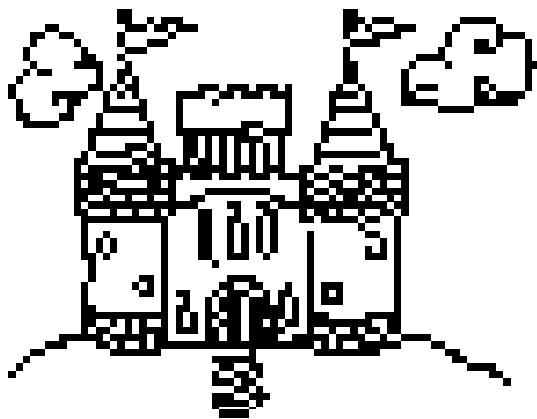
# Database Security Today – A Brief Overview

- Tedious hardening procedures of OS platform including many vendor specific issues
  - Requires a system administrator
- Careful configuration of esoteric options within database server
  - Requires a database administrator
- Proper coding of applications
  - Requires the good will and attention of programmers
- Tedious object and system privileges management within the database
  - Requires a database administrator working in tandem with the application programmer

# Pitfalls of Current Database Protection Approaches

- The worst “worst case” assumption
  - Anyone can try anything (an omnipotent attacker)
  - The omnipotent attacker can be anywhere
  - The omnipotent attacker will try every possible attack whenever it is possible
- Protect anything using any available technique
- Based on misinterpretation of the Attacker – Defender asymmetric effort problem:
  - An attacker need only find a single vulnerability while a defender must counter all possible threats.

# Demo



How many guards are required for our castle?

# Pitfalls of Current Database Protection Approaches

- Consequences:
  - Dispersion of resources
  - Over budgeting
  - All attackers get the same (very limited) attention

# Pitfalls of Current Database Protection Approaches

- Demand for Tight Integration Between Application Programmer, Database Administrator, Security Officer
  - Granular ACLs
  - Multiple accounts
- Integrate Security Requirements into Capacity Planning:
  - How much additional CPU power is required from the database server machine if ACLs are applied or if audit is turned on?
  - How much additional disk space and I/O bandwidth is required?
  - What is the cost of such additional resources for high-end database servers?

# Pitfalls of Current Database Protection Approaches

- Contradicting Requirements:
  - Use “Prepared Statements” to access the database
  - Audit all access attempts to sensitive tables (will audit only “Prepare” command and not “Execute”).
- Technical Shortcomings
  - Built-in audit capabilities do not audit values (requires building triggers for each table)
  - Built-in audit ACLs do not enforce specific queries (requires the use of views or stored procedures)

# Database security threats revisited – Client Based Classification Scheme

- Attackers are not into academic research hence:
  - An attacker will use the simplest available methods (this is especially true for most non-criminal attackers) before resorting to complex ones.
  - An attacker will employ a given technique only if it provides additional access (e.g. a user with DBA privileges will not use brute-force login attacks)
  - Some attackers may have to go through various unrelated security mechanisms before they are in a position to attack the database (e.g. a hypothetical internal database with no outside network connections requires a physical security breach before it can be accessed by unauthorized personnel)

# Database security threats revisited – Client Based Classification Scheme

- By analyzing the attacker's situation we can achieve the following:
  - Better balance between protection within the database and other protection mechanisms.
  - Focused efforts on effective countermeasures against actual threats
- Counter Example (the worst “worst case” assumption):
  - Given a database server connected to a web server
  - “Let's protect the web application as though the database server is unprotected and protect the database as though the application is completely insecure”
  - Never enough resources

# Database security threats revisited – Client Based Classification Scheme

- As a consequence not only that we achieve a more effective protection scheme but we simplify security management in the database, further contributing to the overall security score.
- An example: Given a database server connected to a web server. An attacker from the Internet cannot (and will not) perform login brute-forcing given that the perimeter FW is properly configured. Hence the amount of resources we put into protecting against it should be proportional
- Claim:
  - If we analyze the attacker's characteristics as a client of the database server we obtain a better understanding of threats and countermeasures

# Database security threats revisited – Client Based Classification Scheme

- Client based attack classification comprises the following stages:
  - Identifying the various types of client groups that use the database.
  - Analyzing the characteristics of attackers in each of the groups in terms of interfaces, capabilities and probable attacks
  - Articulate the most effective countermeasures required to thwart attackers from within each group

# Client Based Attack Classification – Main Client Group Types

- Internet Attackers
  - Many malicious individuals with indirect access through an application (usually a web application).
  - Attacks rely on technical IT skills.
- Intranet Attackers
  - Some malicious individuals with indirect access through 3 tier applications (web applications or traditional client server applications)
  - Most “attacks” by internal users (embezzlement, data theft, etc.) do not involve any technical IT skills.

# Client Based Attack Classification – Main Client Group Types

- Thick Clients
  - Direct access to the database server using the SQL query language
- Non-Users
- Administrative Users

# Client Based Attack Classification – Client Characteristics

	Internet Users	Intranet Users	Thick Clients
Network Access – DB Protocol	Indirect	Indirect	Direct
Network Access – Other Protocols	None	Unintentional	Unintentional
Source IP Address	Few, Static	Few, Static	Many, Dynamic
Connection Policy	Multiple, Pooled	Multiple Pooled	Single

# Client Based Attack Classification – Client Characteristics (Cont.)

	Internet Users	Intranet Users	Thick Clients
Database Accounts	Few	Few	Few (occasionally many)
Login Method	Automatic	Automatic	Automatic
SQL Access	Limited Set, No DDL	Limited Set, No DDL	Limited Set, No DDL
Attacker Skills	High Technical	Low Technical, High Professional	Low Technical, High Professional

# Client Based Attack Classification – Client Characteristics (Cont.)

	Internet Users	Intranet Users	Thick Clients
Attack Environment	Comfortable	Uncomfortable	Uncomfortable
Covering Tracks	Easy	Difficult	Very Difficult

# Client Based Attack Classification – Client Characteristics (Cont.)

	Administrators	Non Users
Network Access – DB Protocol	Direct	Yes
Network Access – Other Protocols	Direct (NetBios, SNMP)	Unintentional
Source IP Address	Few Dynamic and Few Static	Many, Dynamic
Connection Policy	Single	None

# Client Based Attack Classification – Client Characteristics (Cont.)

	Administrators	Non Users
Database Accounts	Few Individual or Shared	None
Login Method	Manual	Automatic
SQL Access	Not Limited	None
Attacker Skills	High Technical	High Technical

# Client Based Attack Classification – Client Characteristics (Cont.)

	Administrators	Non Users
Attack Environment	Uncomfortable	Uncomfortable
Covering Tracks	Easy	Very Difficult

# Client Based Attack Classification – Internet Attackers

- Most Probable Attacks
  - SQL Injection (SQL statements that are an extension to known statements)
- Other Possible Attacks
  - Compromise the web server and launch network level attacks
  - Compromise the web server and launch an SQL client using the application credentials
- Improbable Attacks
  - Compromise the web server and launch brute force login attacks on database

# Client Based Attack Classification – Internet Attackers

- Suggested Countermeasures

- Restrict network access between application and database to the database protocol (SQL\*NET, TDS, DRDA, etc.) and to the known addresses of the web servers
- Use a dedicated account for each application. Restrict access from the IP address of web servers to the given application account (and allow access to the account only from web server addresses)
- Restrict access of the given application account to the set of known queries (e.g. an application may invoke "*select user\_id from app\_users where user\_name = ? And password = ?*" but is not allowed to invoke "*select user\_name, password from app\_users*")

# Client Based Attack Classification – Internet Attackers

- Suggested Countermeasures (Cont.)
  - Enforce a zero-tolerance failed login policy:
    - On source IP addresses (Failed login attempt = compromised web server)
    - On given account
  - Secure Web Server
  - Inspect database protocol traffic with IDS / IPS

# Client Based Attack Classification – Countermeasures

	Internet Attackers	Intranet Attackers	Thick Client Attacks
Network ACL	DB protocol only from Web Servers	DB protocol only from Application Servers	DB protocol only
IPS / IDS	On DB protocol	On DB Segment	On DB Segment
Audit		Granular	Granular
External Measures	Secure Web Server	Secure App Server	

# Client Based Attack Classification – Countermeasures (Cont.)

	Internet Attackers	Intranet Attackers	Thick Client Attacks
DB Connection Control	Application Account  Account to IP restrictions  0 Tolerance on Failed Logins	Application Account  Account to IP restrictions  0 Tolerance on Failed Logins  Time of Day	0 Tolerance on Failed Logins  Time of Day  Single connection per IP
SQL Statement Level Control	Restrict to known set  Deny DDL	Restrict to known set  Deny DDL	Restrict to known set  Deny DDL

# Client Based Attack Classification – Countermeasures (Cont.)

	Administrators	Non Users
Network ACL	Administrative servers VPN	DB protocol only
IPS / IDS	On DB Segment	On DB Segment
Audit	Granular Independent	Granular
External Measures	Data Encryption Physical Redundancy	

# Client Based Attack Classification – Countermeasures (Cont.)

	Administrators	Non Users
DB Connection Control	Individual Account Failed login policy	Default Accounts Failed login policy
SQL Statement Level Control		

# Client Based Attack Classification – Intranet Attackers

- Most Probable Attacks
  - SQL Injection
  - Privilege Abuse (e.g. Copy all medical records to a memory stick)
- Other Possible Attacks
  - Compromise the web server and launch network level attacks
  - Launch network level attacks expecting internal network access controls to be loose (requires knowledge of the database server's IP address).
  - Launch login brute force attack against the database, expecting internal network controls to be loose

# Client Based Attack Classification – Intranet Attackers

- Improbable Attacks
  - Compromise the web server and launch brute force login attacks on database
  - Launch complex network based attacks

# Client Based Attack Classification – Intranet Attackers

- Suggested Countermeasures

- Restrict network access between application and database to the database protocol and to the known addresses of application or web servers
- Use a dedicated account for each application. Restrict access from the IP address of application or web servers to the given application account (and vice-versa)
- Restrict access of the given application account to the set of known queries (e.g. an application may invoke *"select user\_id from app\_users where user\_name = ? And password = ?"* but is not allowed to invoke *"select user\_name, password from app\_users"*)

# Client Based Attack Classification – Intranet Attackers

- Suggested Countermeasures (Cont.)
  - Enforce a zero-tolerance failed login policy:
    - On source IP addresses (Failed login attempt = compromised application server)
    - On given account
  - Enforce “Time of Day” Restrictions (usually applicable)
  - Secure application servers
  - Inspect internal traffic (at least on database segment) using IDS / IPS
  - Audit

# Client Based Attack Classification – Thick Clients

- Most Probable Attacks
  - Use alternative client software (e.g. SQL\*Plus, MS Query Analyzer, etc.)
  - Generate arbitrary database access statements (DDL and DML) using the application's credentials (against non-application tables or against sensitive information)
  - Privilege Abuse (e.g. Copy all medical records to a memory stick)
  - Connect to database using default accounts

# Client Based Attack Classification – Thick Clients

- Other Possible Attacks
  - Brute-force credentials for privilege elevation
  - Launch network level attacks against database communication protocol
  - Launch network level attacks expecting internal network access controls to be loose
- Improbable Attacks
  - Launch complex network level attacks against database server

# Client Based Attack Classification – Thick Clients

- Countermeasures
  - Use a dedicated account for each application role or client software
  - Restrict use of the given application account by the given client software (this is not full proof but fits the profile of most internal attackers)
  - Restrict access of the given application account to the known set of queries
  - Restrict network access between clients and database to the database protocol
  - Restrict number of concurrent network connections from each source IP to the database

# Client Based Attack Classification – Thick Clients

- Countermeasures (Cont.):
  - Enforce a zero-tolerance failed login policy on given account. Pay special attention to default accounts (regardless of whether they actually exist in your database or not)
  - Enforce “Time of Day” Restrictions (usually applicable)
  - Inspect internal traffic (at least on database segment) using IDS / IPS
  - Audit

# Client Based Attack Classification – Non Users

- Most Probable Attacks
  - Launch simple network attacks as part of a non-targeted attack
  - Launch simple and complex network attacks targeted on the database server and probably the database protocol
  - Use of default accounts or brute forcing database credentials

# Client Based Attack Classification – Non Users

- Countermeasures:
  - Restrict network access to database servers to database protocol only.
  - Inspect internal traffic (at least on database segment) using IDS / IPS
  - Detect the use of (otherwise unused) default accounts
  - Detect login brute forcing (same IP, **many failed attempts**)
  - Detect the use of (otherwise unused) client software

# Client Based Attack Classification – Administrative Users

- The aim is to positively identify administrative users rather than try and stop them
- Administrative users include both database and system administrators both groups have equal damaging potential in all aspects

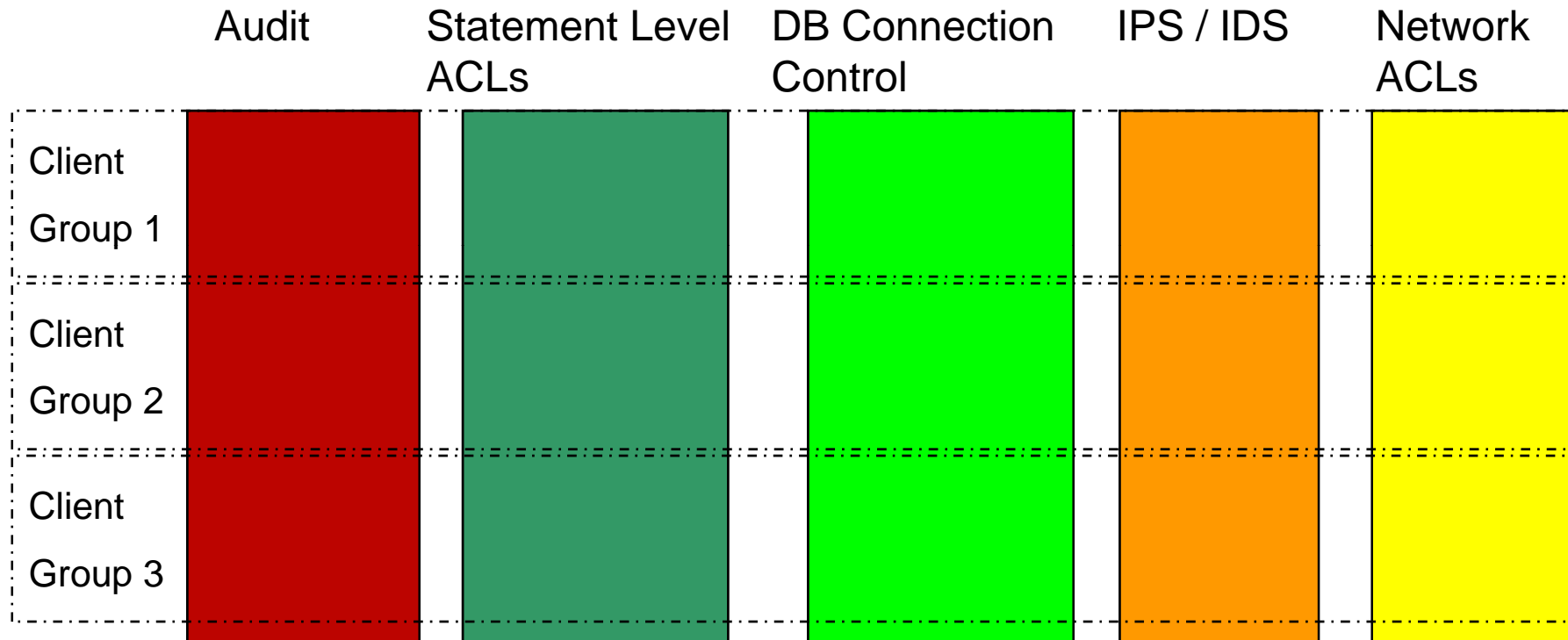
# Client Based Attack Classification – Administrative Users

- Probable Attacks
  - Privilege abuse
  - Data destruction
  - Confidentiality breach (when administrative personnel should not be exposed to data)

# Client Based Attack Classification – Administrative Users

- Countermeasures:
  - Require individual identification
  - Require strong authentication methods on an administrative network connection
  - Audit – it is important that audit is performed by a mechanism that is not controlled by the administrator
  - Physical data redundancy
  - Data encryption by application

# Constructing an Effective Solution



# Constructing an Effective Solution

- Identify and specify all client groups
- Derive from the client group specification the configuration required for all enforcement layers:
  - Network ACLs (Firewall)
  - IDS / IPS
  - Database Connection Control
  - Statement Level ACLs
  - Audit

# Constructing an Effective Solution – Client Group Specification

- Generic Client Group Description
  - Type (Application Server, Thick Client, Administrative, RoW)
  - Source IP addresses
  - Client software
  - Database account
  - Set of SQL statements
  - Additional network protocols

# Constructing an Effective Solution – Client Group Specification

- Sample Application Server Group
  - Source IP addresses: 192.168.1.1, 192.168.1.2
  - Client software:
  - Database account: app\_user1
  - Set of SQL Statements:
    - Select user\_id from app\_users where ....
    - Insert into orders values .....
    - Select \* from orders where ....
  - Additional network protocols: None

# Constructing an Effective Solution – Client Group Specification

- Sample Administrative Server
  - Source IP addresses: 192.168.10.1
  - Client software: None
  - Database account: None
  - Set of SQL statements: None
  - Additional network protocols: SNMP, ICMP Ping

# Constructing an Effective Solution – Client Group Specification

- How to identify and specify the set of SQL statements?
  - How would you identify the set of required database privileges?
- Solution #1: Ask Programmers
  - They don't know until they finished coding
  - When they finish coding they have new stuff to code and have no record of their previous work
- Solution #2: Monitor Application Traffic

# Constructing an Effective Solution – Client Group Specification

- Specifying all allowed statements can be tedious (and sometimes impossible)
  - Use a mixture of strict specifications together with wild-cards
  - Can start with all wild-cards and improve over time
  - Strict:
    - Select \* from orders where customer = ? Order by date desc
  - Wild-card:
    - Select \* from orders
    - All DDL commands
    - All DML commands on orderItems

# Constructing an Effective Solution

- Network ACLs
  - Can use any simple Firewall
  - May implement VPN for secure administrative connections
- IPS / IDS
  - Should possess generic capabilities (e.g. Snort™ compliant)
  - Should have strong capabilities with regard to database network protocols

# Constructing an Effective Solution

- Database Connection Control
  - Decisions based on Source IP address, Target Account Client Software
  - Enforced policies may include:
    - Access control
    - Failed logins
    - Number of concurrent connections
    - Time based controls (time of day, life span of a connection, etc.)

# Constructing an Effective Solution

- Statement Level ACLs
  - Can be degraded to simple object ACLs using wild-cards
  - Delivers the true ability to control access to data in a real world scenarios:
    - 3 tier applications
    - Multiple applications using the same set of database tables

# Constructing an Effective Solution

- Accurate Audit
  - It's not enough to know that some application issued a "Select" statement against the credit cards table. We want to see the parameters used for the query
- Independent Audit
  - An audit mechanism that does not affect the performance and behavior of the database server
  - An audit mechanism that cannot be tampered using built-in database server capabilities and in particular cannot be tampered by administrative database users

# Summary

- Classifying our environment into client groups or potential attacker groups allows us to define and design a simple yet effective model for protecting a database server (or a data center for that matter).
- While Network ACLs and IPS / IDS layers are currently under control of security officer, other layers are usually considered to be the task of database administrators:
  - No direct control of security officer
  - Implementation and the level of implementation highly depends on the specific database vendor
  - Implementation is usually complex and costly
  - Security mechanisms rely on the secured object (paradox)

# Summary

- We need a new breed of security technologies that will put the power to implement database security, according to this model, back in the hands of the security officer.
- Such technologies should be
  - Independent of specific database platform
  - Independent of the database server itself
- The new technologies should strive to implement all required database protection layers within a single package.



---

# Thank You

Imperva, Inc.

3400 Bridge Parkway, Suite 101, Redwood Shores CA 94065

Sales: +1-866-926-4678 [www.imperva.com](http://www.imperva.com)

