Policy per la sicurezza dei Database

Danilo Massa
IT Security Expertise Center Leader
GIAC SSP-GHD, GWAS, GCIH, GREM, GCFA Certified Professional

Sessione di studio AIEA - Torino, 3 luglio 2008

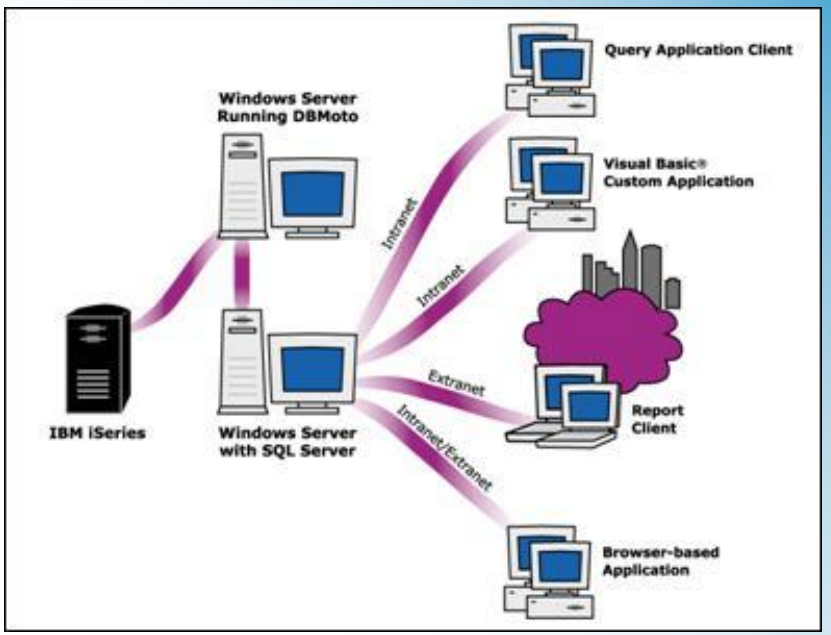
Database – Policy, Sicurezza e Hardening

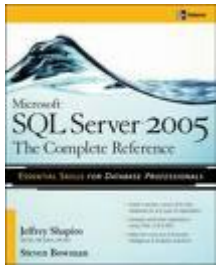
1. Sicurezza dei DB
2. Elementi di Sicurezza comuni per i DB
3. Elementi di Sicurezza Applicativa
4. Elementi aggiuntivi di Sicurezza
5. Procedure per l'Hardening dei DB

Sicurezza dei DB [1/3]

Oggigiorno, larghissima diffusione dei Database …

… quasi il 100% delle applicazioni Web e Client/Server si appoggia su un DB …

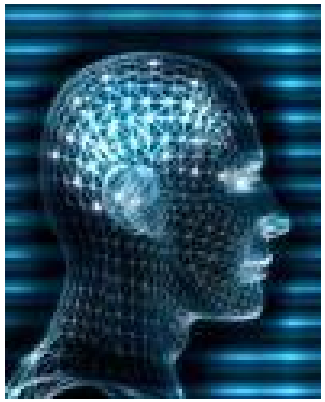




Analizzeremo quindi le policy più importanti relative alla sicurezza delle BASI DATI, senza fare riferimento ad applicazioni/vendor specifici

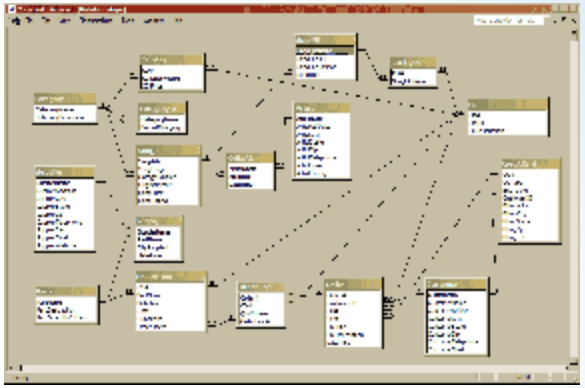
Sicurezza dei DB [2/3]

Primo step FONDAMENTALE:



PENSARE IN SICUREZZA

già dal design del database …



+

scegliere un RDBMS che permetta una semplice ed efficace implementazione delle policy e delle componenti di sicurezza studiate a priori

Sicurezza dei DB [3/3]

Prima di poter applicare eventuali **policy di sicurezza per i DB**, è necessario che ogni singola applicazione che li utilizza sia catalogata e ben descritta

Utilizzo di **strumenti automatici** per la gestione degli asset + **raccolta informazioni**

Lo scopo è catalogare:

- Le procedure di integrazione applicativa (quale base dati è utilizzata da un certo application server)
- La catalogazione della riservatezza dei dati
- Le attività di gestione ordinaria e straordinaria dei sistemi/applicativi
- Le figure che sono responsabili per il corretto funzionamento delle applicazioni o dei sistemi
- Le procedure per la gestione degli incidenti

Elementi di Sicurezza comuni per i DB [1/16]

Autenticazione

I database forniscono solitamente un loro sistema di autenticazione, oppure sono in grado di appoggiarsi ad altri sistemi di autenticazione esterna (es. Server LDAP, AD, token, smart card …)

In questo caso, le applicazioni possono utilizzare uno dei seguenti metodi per l'autenticazione delle utenze:

1. Utilizzo di un utente RDBMS
2. Utilizzo dell'autenticazione da sistema operativo
3. Utilizzo di una tabella autenticativa custom

Elementi di Sicurezza comuni per i DB [2/16]

Autenticazione

I - Utilizzo di un utente RDBMS

PRO

- Le password degli utenti sono già memorizzate in modalità sicura
- E' possibile limitare gli utenti permettendo operazioni soltanto su tabelle/viste ben definite
- Eventuali vulnerabilità legate alla gestione degli utenti sono corrette dal produttore e testate da un elevato numero di utilizzatori

CONS

- Alcuni RDBMS hanno il costo che è funzione diretta degli utenti registrati sullo stesso
- Aumento della complessità applicativa per la gestione degli utenti e dei ruoli
- Necessità di utenti specifici per l'esecuzione di attività di audit e/o batch

Elementi di Sicurezza comuni per i DB [3/16]

Autenticazione

I - Utilizzo di un utente RDBMS

Consigli per la policy

Verificare l'utilizzo di utenti di database

Verificare o definire una procedura automatica (es. via batch o meglio con un Identity Manager) per la verifica della congruenza tra gli utenti censiti sul database e quelli che debbono effettivamente accederci (es. non vi siano persone che si sono dimesse o che sono state trasferite ad altra unità lavorativa che non necessita accesso all'applicazione)

Verificare la profilazione degli utenti in modo che i permessi di accesso siano limitati alle sole tabelle/viste necessarie e non sia possibile accedere ad altri DB e/o tabelle di sistema

Elementi di Sicurezza comuni per i DB [4/16]

Autenticazione

II – Autenticazione da Sistema Operativo

E' la soluzione MIGLIORE

PRO

- Tutti i vantaggi della soluzione precedente (utilizzo di un utente RDBMS)
- Soluzione integrata nell'ambiente operativo: non richiede il transito sulla rete delle coppie login/password necessarie affinché l'autenticazione abbia luogo

CONS

- Possibile aumento della complessità applicativa per la gestione degli utenti e dei ruoli definiti per gli utenti nel sistema operativo, in relazione all'utilizzo degli stessi all'interno del RDBMS

Elementi di Sicurezza comuni per i DB [5/16]

Autenticazione

II – Autenticazione da Sistema Operativo

Consigli per la policy

Verificare l'utilizzo di utenti di sistema operativo

Verificare o definire una procedura automatica (es. via batch o meglio con un Identity Manager) per la verifica della congruenza tra gli utenti censiti sul database e quelli che debbono effettivamente accederci

Verificare la profilazione degli utenti in modo che i permessi di accesso siano limitati alle sole tabelle/viste necessarie e non sia possibile accedere ad altri DB e/o tabelle di sistema

Verificare le modalità di autenticazione da remoto (cioè fuori dalla rete locale, se previste)

Elementi di Sicurezza comuni per i DB [6/16]

Autenticazione

III – Uso di tabella autenticativa Custom

E' la soluzione più pericolosa

Sicurezza delle credenziali in gestione agli sviluppatori

Problematiche Comuni

- Password degli utenti non cifrate nella tabella di autenticazione
- Non vi possono essere discriminazioni agli accessi sulle tabelle in base agli utenti
- Molto spesso l'applicazione si autentica sulla base dati utilizzando credenziali riservate ai DBA (es. system per Oracle, sa per MS SQL Server, root per MySQL)
- Gli accessi non controllati dall'applicazione (es. via ODBC) avvengono con le credenziali dell'applicazione e quindi con possibilità elevate di causare danni
- Nel caso in cui l'applicazione sia vulnerabile a qualche tipo di attacco (es. SQL Injection) è molto semplice per un hacker ottenere l'elenco di tutti gli utenti e le loro password

Elementi di Sicurezza comuni per i DB [7/16]

Autenticazione

III – Uso di tabella autenticativa Custom

Tale soluzione può essere utilizzata in casi particolari:

- Applicazioni di terze parti leader di mercato o di rilevanza elevata per il business
- Applicazioni datate per cui non è possibile eseguire delle modifiche
- Applicazioni per cui sia stata effettuata una completa verifica di sicurezza applicativa e/o una formale code review

Elementi di Sicurezza comuni per i DB [8/16]

Autenticazione

III – Uso di tabella autenticativa Custom

Consigli per la policy

Verificare l'utilizzo di password cifrate nella tabella che autentica gli utenti

Verificare che l'applicazione non utilizzi un utente amministratore di base dati (es. system, sa, root) per operare

Verificare la congruenza tra gli utenti censiti sulla tabella autenticativa e quelli che debbono effettivamente accederci (es. non vi siano persone che si sono dimesse o che sono state trasferite ad altra unità lavorativa che non necessita accesso all'applicazione)

Eseguire una completa analisi di sicurezza sull'applicazione con la verifica delle utenze definite

Elementi di Sicurezza comuni per i DB [9/16]

Autorizzazione

Consente la gestione dei privilegi assegnati agli utenti

La maggior parte dei prodotti di RDMBS permette di specificare per ogni utente o profilo, quali sono i suoi **privilegi CRUD** (create, read, update e delete) in modalità molto fine (tabelle, viste etc)

Inoltre è anche possibile **specificare i permessi di esecuzione delle stored procedure**.

Viene specificata la "libertà" che l'utente ha di spaziare all'interno del DB

Elementi di Sicurezza comuni per i DB [10/16]

Autorizzazione

2 Diverse possibilità implementative:

Utilizzo delle funzionalità native del RDBMS

Utilizzo di tabelle autorizzative custom

Consigli per le policy

Verificare che ogni utente sia correttamente profilato con i minimi permessi necessari per l'esecuzione delle funzionalità applicative

Verificare che l'utente possa eseguire soltanto le stored procedure necessarie per il funzionamento dell'applicazione (non quelle di sistema)

Verificare la congruenza tra gli utenti censiti sulla tabella autorizzativa ed il loro profilo da utilizzatore

analisi di sicurezza sull'applicazione + code review per verificare l'impossibilità di eseguire escalation dei privilegi

Elementi di Sicurezza comuni per i DB [11/16]

Confidenzialità

RDBMS permettono:

- Creazione di tabelle con contenuto crittato
- Utilizzo di canali di comunicazione client/server cifrati

Per elevarne la sicurezza, alcuni permettono:

Crittazione del codice applicativo in esso contenuto (stored procedure)

E' necessario utilizzare la funzionalità di cifratura offerta da un database soltanto in alcuni casi specifici

Dati Segreti

Informazioni Sensibili

Perché ci sono alcuni inconvenienti …

Elementi di Sicurezza comuni per i DB [12/16]

Confidenzialità

Aspetti negativi della cifratura:

- Difficoltà di debug applicativo (non è possibile verificare in modo semplice i dati letti o scritti dall'applicazione)
- Difficoltà di recovery nel caso in cui non sia possibile recuperare i dati persi da un backup
- Aumento delle risorse macchina necessarie per la computazione (problemi di performance)

Consigli per la policy

Verificare che tutti i dati ritenuti altamente riservati, segreti, o sensibili siano memorizzati in modalità cifrata sulla base dati

Verificare che le stored procedure utilizzate per il trattamento di dati altamente riservati, segreti o sensibili siano memorizzate in modalità cifrata sulla base dati

Elementi di Sicurezza comuni per i DB [13/16]

Integrità

Per rendere più sicura e “integra” una base dati, è opportuno fare uso delle funzioni di **integrità referenziale** che … molto spesso non sono invece utilizzate …

E’ importante usare tali funzionalità per garantire l’integrità dei dati presenti sul database sia rispetto ad una *singola applicazione*, sia nel caso in cui sulla stessa base dati insistano *più applicazioni diverse*.

Integrità Referenziale: NUMEROSI VANTAGGI !!!!

Un solo svantaggio:

aumento della complessità in fase di progettazione e sviluppo

Elementi di Sicurezza comuni per i DB [14/16]

Integrità

Consigli per le policy

Verificare l'utilizzo dei sistemi di integrità referenziale della base dati

Verificare che l'applicazione gestisca correttamente le transazioni sulla base dati e non vi siano dati inconsistenti

Elementi di Sicurezza comuni per i DB [15/16]

Audit

Utile per la sicurezza

+

per il debug applicativo

+

per effettuare l'hardening di una base dati già in produzione

Diversi RDBMS permettono di attivare funzionalità di audit per l'esecuzione di operazioni specifiche o privilegiate, registrando all'interno di un apposito file di log tali attività

Logging abilitato almeno negli ambienti di PRODUZIONE e TEST

+

Backup dei file di LOG

Elementi di Sicurezza comuni per i DB [16/16]

Audit

Consigli per le policy

Verificare la sincronizzazione del sistema operativo con un time server per eventuali correlazioni dei log di audit tra la base dati, l'application server ed altri componenti dell'architettura (es. firewall)

Verificare l'attivazione, il funzionamento ed il corretto backup del log di audit della base dati

Verificare l'attivazione, il funzionamento ed il corretto backup del log di audit dell'applicazione

Elementi di Sicurezza Applicativa [1/6]

Gestione delle credenziali di accesso per **applicazioni**

3 Metodologie e loro Sicurezza:

1. Richiesta delle credenziali all''utente (uso di un utente RDBMS)

L'pplicazione richiede le credenziali all'utente e le usa per accedere alla base dati. Lato Security, è necessario solamente verificare che la coppia user e password transiti cifrata sulla rete

2. Autenticazione da file

L'applicazione dovrà utilizzare una coppia user e password memorizzata da qualche parte. Lato Security, è necessario verificare:

- *Le credenziali di accesso siano memorizzate cifrate con un algoritmo sicuro*
- *Il file o il repository che contiene le credenziali non sia accedibile da persone non autorizzate*
- *Le credenziali non siano memorizzate in chiaro in alcun tipo di file*

3. Autenticazione da sistema operativo

Situazione Ottimale: *direttamente il database a richiedere le credenziali di autenticazione utente al sistema operativo. Osservazione lato Security: coppia user password non transita sulla rete.*

Elementi di Sicurezza Applicativa [2/6]

Gestione delle credenziali di accesso per **applicazioni**

Consigli per le policy

Verificare che le credenziali non siano passate in chiaro durante la connessione con la base dati

Nel caso in cui risulti necessario memorizzare le credenziali di accesso alla base dati, verificare che queste siano cifrate nel file o repository che le contiene e che non sia accedibile da personale non autorizzato

Verificare che nel codice applicativo (compilato o meno) non compaiano le credenziali di accesso in chiaro

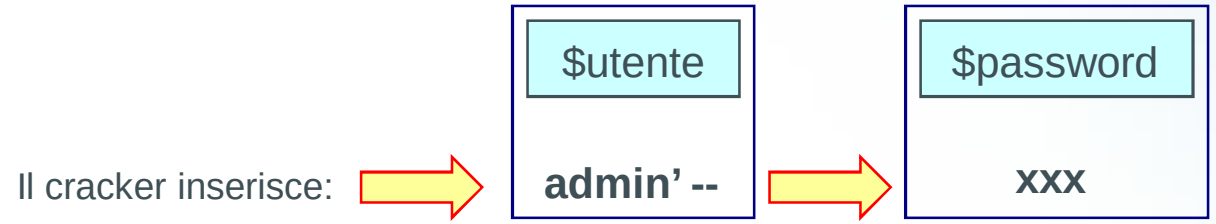
Elementi di Sicurezza Applicativa [3/6]

Protezione da attacchi di SQL Injection

E' una TECNICA COMUNE di attacco ai DB

SQL Injection: consiste nel tentare l'*esecuzione di SQL non autorizzato inserendo nei campi dell'applicazione stringhe che alterano il codice in esecuzione*

Scenario: *applicazione con due campi da compilare per eseguire l'accesso autenticato ad una tabella del DB:*



Se il codice applicativo, per eseguire l'autenticazione, utilizza la query SQL:

SELECT id FROM tabella_utenti WHERE user='$utente' AND password='$password'

Inserendo Utenza e Password sopra definite, otterremmo la seguente query:

SELECT id FROM tabella_utenti WHERE user='admin' --' AND password='xxx'

Elementi di Sicurezza Applicativa [4/6]

Protezione da attacchi di SQL Injection

SELECT id FROM tabella_utenti WHERE user='admin' --' AND password='xxx'

Tale query ricerca solamente un utente e lo autentica indipendentemente dalla password

Per scongiurare tali tipi di attacchi, è necessario eseguire dei controlli specifici lato-server

Controlli su eventuali dati inseriti da un utente

Controlli di qualsiasi tipo di dato che l'applicazione accetta dall'esterno

(file da importare, parametri ricevuti da web service etc.)

Tipi di Controlli:

- Controllare tutti i caratteri ricevuti in input ed eliminare quelli non concessi
- Sostituire i caratteri pericolosi con gli equivalenti quotati (es. il singolo apice deve essere ripetuto)
- Forzare o non accettare dati che non rispettino il formato atteso in input

Elementi di Sicurezza Applicativa [5/6]

DBLink

Alcuni RDBMS permettono la creazione di “link” tra diverse istanze di base dati, istanze che possono essere sia locali (sulla stessa macchina) sia remote (su macchine diverse)

E’ necessario porre attenzione nell’uso di questa caratteristica e verificarne la reale necessità al momento della progettazione applicativa.

Problematiche Comuni

- L’integrità referenziale dei dati può essere difficoltosa se eseguita tra due basi dati diverse (e quindi rischia di non essere implementata)
- La compromissione di una base dati da parte di un hacker può comportare rischi per la base dati collegata

Elementi di Sicurezza Applicativa [6/6]

DBLink

Consigli per le policy

Verificare le modalità di mantenimento dell'integrità dei dati (solitamente implementate applicativamente)

Verificare le modalità di memorizzazione delle credenziali utilizzate per la creazione del link (queste non devono essere rese leggibili o modificabili da utenti non amministratori della base dati)

Verificare che le credenziali di autenticazione fra le basi dati siano ristrette al solo DBLink

Elementi aggiuntivi di Sicurezza [1/10]

Installazione RDBMS

Spesso, le installazioni standard di RDBMS eseguono delle installazioni di default sul sistema:

- Utenti amministrativi di default (es system/manager su Oracle)
- Database di esempio (es. test su MySQL)
- Utenti di esempio (es. scott/tiger su Oracle)
- Componenti aggiuntive per la gestione o gli sviluppi applicativi

E' necessario prestare particolare attenzione a queste componenti !!!!

In particolare, è necessario che il **DBA** assegnato *effettui tutte le verifiche necessarie alla corretta configurazione della base dati* che sarà utilizzata e che **tutte le caratteristiche o funzionalità non necessarie siano rimosse**, **arrestate** (es. listener di Oracle se non necessario) oppure **correttamente configurate**.

Elementi aggiuntivi di Sicurezza [2/10]

Installazione RDBMS

Consigli per le policy

Verificare la rimozione delle eventuali istanze di esempio

Verificare l'eliminazione degli eventuali utenti di esempio

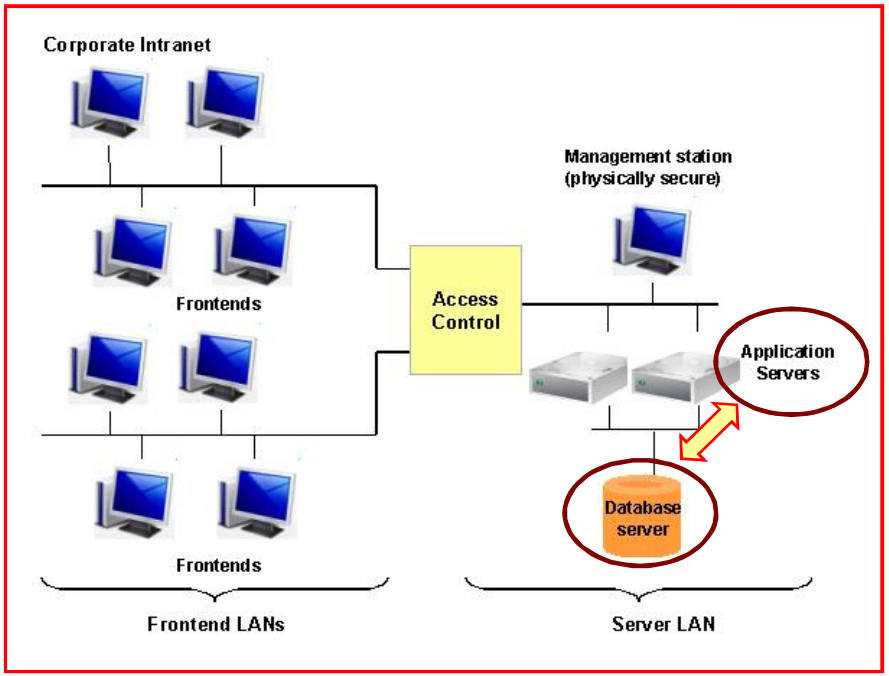
Verificare il cambio password per le utenze di default (es. system su Oracle)

Verificare la corretta configurazione delle componenti di sicurezza (es. audit log)

Elementi aggiuntivi di Sicurezza [3/10]

Networking

Gli **RDBMS sono dotati di listener TCP/IP** che permettono la creazione di architetture client/server a due o tre livelli



In tali architetture la base dati non è installata sulla macchina in cui è invece presente la componente applicativa che la deve utilizzare, e quest'ultima necessita quindi di aprire una comunicazione verso il listener TCP/IP per poter operare correttamente.

tale listener dovrebbe essere solo accessibile dall'application server

Elementi aggiuntivi di Sicurezza [4/10]

Networking

Consigli per le policy

Nel caso in cui l'application server sia installato sulla stessa macchina della base dati, è necessario arrestare il listener del DB oppure filtrarlo mediante firewall locale o regole sul TCP

Nel caso in cui l'application server non sia installato sulla stessa macchina della base dati, limitare l'accesso al listener, in modo che solo l'application server (ed eventuali altri server applicativi) possano accedervi mediante firewall locale o regole sul TCP

Nel caso in cui il software RDBMS permetta di specificare se l'utente è locale oppure remoto verificarne la configurazione (anche se si utilizza un firewall locale o regole di filtro sul TCP)

Elementi aggiuntivi di Sicurezza [5/10]

Processi Batch di Integrazione/Gestione

Spesso in ambito RDBMS si utilizzano script che provvedono ad esportare o importare dati da/su RDBMS

L'utilizzo di tali script è un rischio, poiché:

- Contengono credenziali valide per accedere alla base dati (e molto spesso sono credenziali di alto livello)
- Non sono protetti, a livello di sistema operativo, permettendo a chiunque (o comunque a personale non autorizzato) di modificarli
- Le directory in cui sono memorizzati i file estratti o da importare non sono protette a livello di sistema operativo (permettendo quindi la lettura/scrittura dei dati stessi)

Elementi aggiuntivi di Sicurezza [6/10]

Processi Batch di Integrazione/Gestione

Consigli per ie policy

Utilizzare credenziali con i minimi permessi necessari per l'esecuzione degli script

Proteggere lo script in modo che soltanto le persone autorizzate possano accedervi in lettura/scrittura

Proteggere lo script in modo che soltanto persone autorizzate o strumenti automatici possano eseguirlo

Utilizzare directory diverse per contenere gli script e per ospitare i dati esportati o da importare

Proteggere le directory in cui si trovano i dati esportati o da importare in modo che soltanto persone o strumenti automatici possano accedervi

Elementi aggiuntivi di Sicurezza [7/10]

Monitoring

Se implementati, possono essere anche fonte di elevati problemi di sicurezza, in quanto per poter eseguire alcuni test di verifica (es. accessibilità di una base dati), necessitano di credenziali valide sul servizio da controllare

E' importante considerare anche questa parte, per rendere più sicura tutta l'infrastruttura !!!

Elementi aggiuntivi di Sicurezza [8/10]

Monitoring

Consigli per le policy

Utilizzare credenziali con i minimi permessi necessari per l'esecuzione dei test

Limitare l'accesso al listener del database, in modo che solo il server di monitoring possa accedervi mediante firewall locale o regole sul TCP

Limitare l'accesso agli agenti SNMP eventualmente presenti sulle macchine che ospitano le basi dati in modo che solo il server di monitoring possa accedervi

Verificare l'utilizzo di community non standard oppure dell'utilizzo dell'autenticazione avanzata sugli agenti SNMP

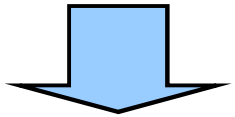
Elementi aggiuntivi di Sicurezza [9/10]

Backup

Necessario per Disaster Recovery

+

Fondamentale verificare la configurazione dei backup …. Ma anche la gestione stessa dei media che contengono le copie di backup deve essere controllata !!!



Tali media non devono essere acceduti da personale NON autorizzato !!!

Elementi aggiuntivi di Sicurezza [10/10]

Backup

Consigli per le policy

Verificare il trattamento e la gestione dei media su cui sono salvate le copie di backup dei dati

Verificare che il sistema di backup non utilizzi copie intermedie (o che queste siano correttamente eliminate al termine del backup)

Nel caso si utilizzino copie di basi dati sincronizzate per l'esecuzione dei backup, queste non devono essere accedibili da personale non autorizzato

Procedure per l'Hardening dei DB [1/5]

Definizione delle Policy

Occorre definirle in modo adeguato + DIVULGARLE !!!

In tale ambito, sono necessari *strumenti tecnici che aiutino la gestione e la verifica della corretta implementazione delle policy da applicare*, per rendere più efficace la gestione della sicurezza

Le indicazioni relative alla sicurezza delle basi dati contenute nelle slides precedenti devono essere estese per coprire particolarità e funzionalità specifiche per ogni RDBMS utilizzato (es. Oracle, MS SQL, MySQL etc).

Procedure per l'Hardening dei DB [2/5]

Definizione delle Policy

Nella definizione delle policy è necessario:

- Documentare correttamente tutte le componenti infrastrutturali ed applicative
- Valutare i rischi associati alla riservatezza dei dati, ovvero catalogare ogni policy in base alla necessità di implementazione rispetto al valore dei dati da proteggere
- Identificare modalità e tempi per l'esecuzione di audit sulla corretta implementazione delle policy (es. definizione di audit interni periodici)
- Individuare strumenti tecnici di controllo sull'implementazione delle policy per evidenziare in tempi ridotti eventuali problemi di sicurezza (es. utilizzare strumenti HIDS per la verifica delle configurazioni)

Procedure per l'Hardening dei DB [3/5]

Diffusione delle Policy

E le modalità di diffusione devono far parte delle policy stesse …

Il primo importante passo, è diffondere le policy di sicurezza **J**

Ma è altresì importante che tali policy siano:

- **applicate**
- **ricordate dagli interessati**

Metodi di efficiente diffusione:

- Organizzare seminari di presentazione delle policy per tutti gli interessati
- Pubblicare in modo facilmente accessibile le policy stesse (es. portale interno)
- Definire, soprattutto nel momento della prima diffusione, una forma di supporto diretto gestito da personale competente (es. casella di posta, help desk etc)
- Provvedere con cadenza regolare (3 mesi/6 mesi) ad effettuare un incontro o riunione di aggiornamento delle policy, per verificare eventuali nuove implementazioni o correzioni
- Utilizzare gli audit interni o esterni come momento di verifica ed ulteriore diffusione (es. presentazione dei risultati dell'audit a tutti gli interessati)

Procedure per l'Hardening dei DB [4/5]

Implementazione delle Policy

E' la fase più critica ed importante

In fase di implementazione, è necessario provvedere a:

- documentare tutte le infrastrutture (solitamente server) che compongono le architetture da porre sotto policy
- documentare tutte le applicazioni e tutti i flussi di integrazione e gestione che interagiscono con le basi dati
- individuare le policy da adottare basandosi su:
  - livello di riservatezza dei dati
  - impatti sulle applicazioni e sugli utilizzatori
  - impatti sui sistemi di integrazione/gestione
- verificare la documentazione e l'analisi di impatto effettuata "a tavolino" utilizzando strumenti tecnici (es. audit degli accessi alla base dati, sniffer di rete etc.)

Procedure per l'Hardening dei DB [5/5]

Implementazione delle Policy

Solitamente l'implementazione delle policy risulta meno rischiosa se si applica la metodologia del "**divide et impera**"

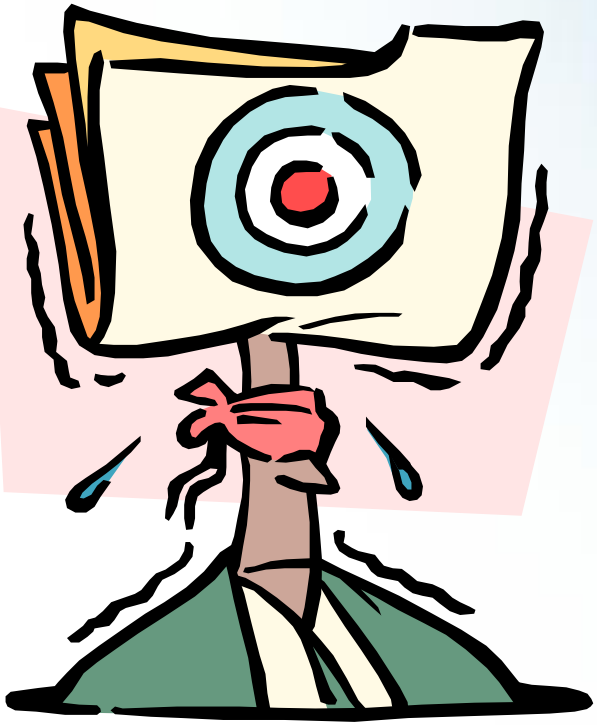
- Catalogazione per importanza (dei dati e per il business)
- Complessità teconologica

→ Definisco sotto-insiemi di basi dati su cui le policy possono essere implementate in modalità graduale

Si opera per piccoli passi in modo graduale … utile per verificare contemporaneamente:

- Funzionamento delle applicazioni che utilizzano i DB
- Funzionamento dei sistemi di integrazione/gestione

Domande e risposte



Domande ?

Danilo Massa
danilo.massa@altran.it

www.altran.com