REGIONE EMILIA-ROMAGNA
Atti amministrativi

GIUNTA REGIONALE

Atto del Dirigente:	DETERMINAZIONE n° 4137 del 28/03/2014
Proposta:	DPG/2014/4203 del 20/03/2014
Struttura proponente:	SERVIZIO SISTEMA INFORMATIVO - INFORMATICO REGIONALE DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA
Oggetto:	DISCIPLINARE TECNICO IN MATERIA DI SICUREZZA DELLE APPLICAZIONI INFORMATICHE NELLA GIUNTA E NELL'ASSEMBLEA LEGISLATIVA DELLA REGIONE EMILIA-ROMAGNA
Autorità emanante:	IL DIRETTORE - DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA
Firmatario:	LORENZO BROCCOLI in qualità di Direttore generale
Luogo di adozione:	BOLOGNA data: 28/03/2014

DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA
IL DIRETTORE

Visto il Decreto Legislativo del 30 giugno 2003, n. 196 "Codice in materia di protezione di dati personali";

Visto il Decreto Legislativo del 7 marzo 2005, n. 82 "Codice dell'amministrazione digitale";

Viste le deliberazioni della Giunta regionale:

- n. 1264 del 01/08/2005 con cui sono state adottate le "Linee guida della Giunta della Regione Emilia-Romagna in materia di protezione dei dati personali";
- n. 2416 del 29/12/2008 "Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali. Adempimenti conseguenti alla delibera 999/2008. Adeguamento e aggiornamento della delibera 450/2007" e in particolare l'Appendice 5 "Trattamento di dati personali con particolare riferimento alla ripartizione di competenze tra i soggetti che effettuano il trattamento";

Viste le determinazioni:

- n. 6928/2009 con cui è stato adottato il "Disciplinare tecnico su modalità e procedure relative alle verifiche di sicurezza sul sistema informativo, ai controlli sull'utilizzo dei beni messi a disposizione dall'Ente per l'attività lavorativa con particolare riferimento alle strumentazioni informatiche e telefoniche ed esemplificazioni di comportamenti per il corretto utilizzo di tali beni, da applicare nella Giunta e nell'Assemblea Legislativa della Regione Emilia-Romagna";
- n. 7222/2012 è stato adottato il "Disciplinare tecnico per la gestione degli incidenti di sicurezza informatica della Giunta e dell'Assemblea Legislativa della Regione Emilia-Romagna"

Dato atto, inoltre, che con propria Determinazione n. 2651/2007 è stato adottato il "Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta";

Considerato che la Regione Emilia-Romagna, nel percorso di informatizzazione e digitalizzazione intrapreso, si propone di

utilizzare sempre nuove risorse e servizi applicativi tecnologicamente avanzati, al fine di raggiungere maggiori livelli di efficienza ed economicità;

Considerato che in questi ultimi anni l'innovazione tecnologica si è notevolmente e velocemente sviluppata e che in virtù della stessa si sono potute approfondire esperienze e conoscenze ulteriori che permettono di individuare nuove misure idonee in materia di sicurezza informatica;

Valutato che, per i motivi sopra esposti, si rende pertanto necessario procedere ad un aggiornamento della policy regionale in materia di sicurezza delle applicazioni informatiche nella Giunta;

Dato altresì atto che saranno redatti ulteriori precisazioni, contenute in Allegati prettamente tecnici e che gli stessi saranno portati a conoscenza, come ulteriori specificazioni del presente atto, tramite diffusione su Internos e tramite comunicazione, anche ai Responsabili esterni fornitori di servizi di sviluppo di applicazioni informatiche;

Vista la determinazione n. 480 del 28/11/2007 "Disciplinare Tecnico in materia di sicurezza delle applicazioni informatiche nella Assemblea legislativa della Regione Emilia-Romagna;

Valutato inoltre che, date le strette connessioni tra i sistemi informatici della Giunta e dell'Assemblea Legislativa e la proficua collaborazione tra i due organi raggiunta in questi ultimi anni, il presente atto sia applicabile anche alle applicazioni informatiche dell'Assemblea Legislativa;

Dato atto che la Cabina di regia ICT di cui alla determinazione n. 2461 del 27 febbraio 2014, è stata informata del contenuto del presente atto;

Acquisito il parere favorevole espresso dal Direttore Generale dell'Assemblea Legislativa, con nota prot. AL 2014.0011615 del 19 marzo 2014;

Sentito il parere del Comitato di Direzione nella seduta del 14 marzo 2014;

Dato atto del parere allegato

d e t e r m i n a

1. di approvare l'allegato "Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta e nell'Assemblea Legislativa della Regione Emilia-Romagna",

comprensivo delle Appendici A, B e C, che ne sono parte integrante e che sostituisce il precedente, adottato con propria Determinazione n. 2651/2007;

2. di applicare all'interno delle proprie strutture l'allegato "Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta e nell'Assemblea Legislativa della Regione Emilia-Romagna";

3. di procedere alla diffusione del contenuto dell'allegato "Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta e nell'Assemblea Legislativa della Regione Emilia-Romagna" a partire dalla data di approvazione del presente atto anche ai Responsabili esterni fornitori di servizi di sviluppo di applicazioni informatiche;

4. di procedere alla diffusione anche degli ulteriori Allegati tecnici che saranno redatti come ulteriori specificazioni del presente atto, tramite diffusione su Internos e tramite comunicazione, anche ai Responsabili esterni fornitori di servizi di sviluppo di applicazioni informatiche.

Lorenzo Broccoli

RegioneEmilia-Romagna

Allegato

Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta e nell'Assemblea Legislativa della Regione Emilia-Romagna

RegioneEmilia-Romagna

INDICE

1. Premessa

Il presente disciplinare descrive gli aspetti tecnici e procedurali richiesti per il design, lo sviluppo, il deployment, il test e la gestione di un'applicazione sicura. Particolare riguardo è dedicato alle applicazioni web, in quanto maggiormente esposte a minacce per la loro caratteristica intrinseca di rendere disponibili servizi ad un numero elevato, e spesso indefinito, di utenti.

2. Applicabilità

Il disciplinare si applica all'interno della Giunta e dell'Assemblea Legislativa della Regione Emilia-Romagna (di seguito denominata Ente) quale strumento di riferimento per i soggetti incaricati di:

- progettare un'applicazione;
- sviluppare un'applicazione;
- acquistare un'applicazione;
- valutare/scegliere fornitori di servizi di sviluppo applicazioni;
- testare la sicurezza di applicazioni;
- adeguare un'applicazione ai criteri di sicurezza previsti dalla normativa vigente;
- installare, gestire o manutenere un'applicazione.

3. Principi generali

3.1 Applicazioni sicure

I destinatari del presente disciplinare devono considerare le minacce di sicurezza e le contromisure disponibili relativamente a dati ed informazioni, secondo le indicazioni fornite nei paragrafi seguenti. Tali indicazioni si basano sul fondamento che un'applicazione è sicura quando è in grado di preservare *riservatezza*/*confidenzialità*, *integrità* e *disponibilità* delle risorse gestite, assicurando costantemente:

- l'identificazione dell'utente che accede alle risorse;
- la limitazione degli accessi alle risorse;
- la comunicazione sicura con l'esterno;
- la conservazione sicura dei dati.

3.2 Architettura applicativa

L'architettura di riferimento di un'applicazione si compone logicamente di tre livelli distinti:

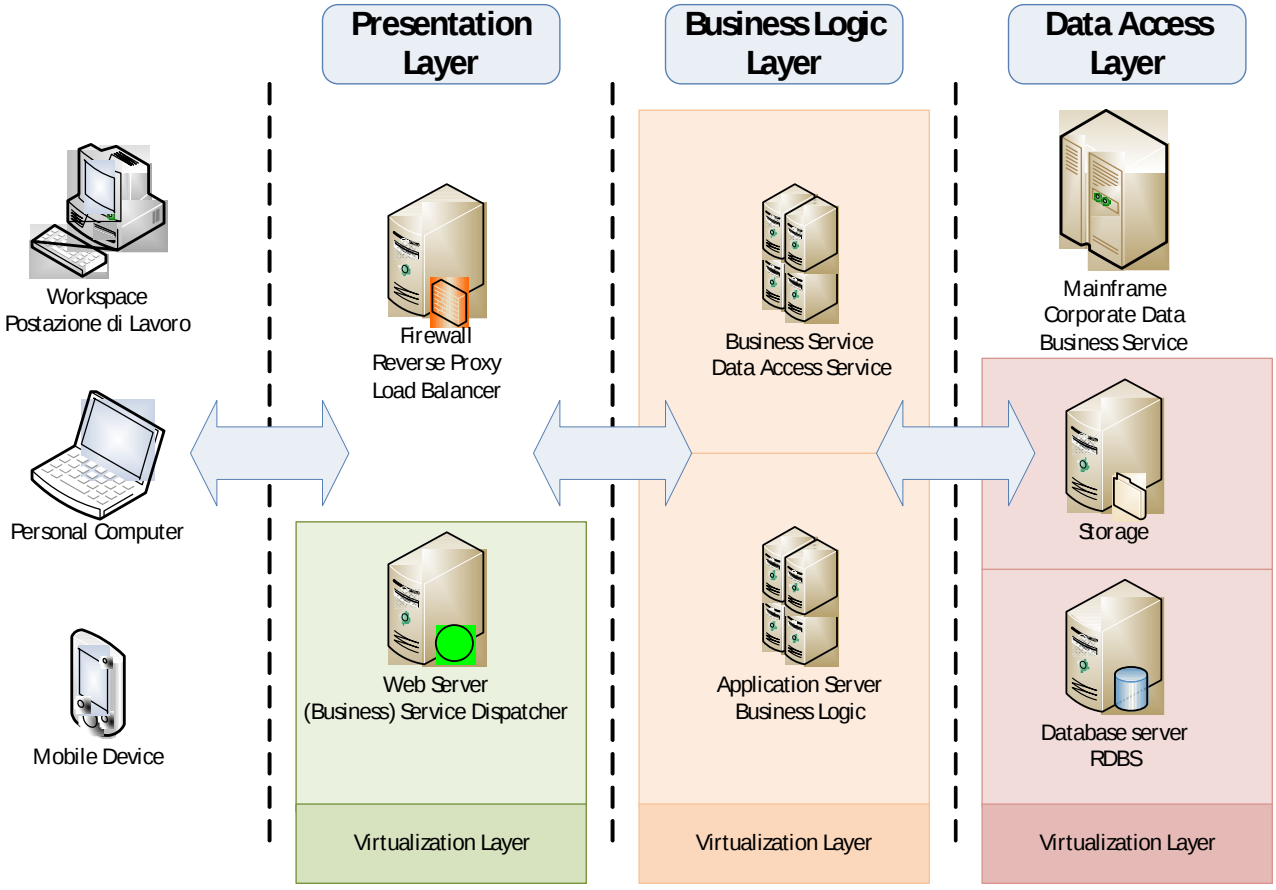
- **livello di presentazione od interfaccia utente**: per la rappresentazione dei dati verso l'utente e della raccolta e verifica dei dati in ingresso;
- **livello business logic od applicazione**: per l'implementazione della logica di

elaborazione dei dati; acquisisce i dati dal livello presentazione e dal livello data access, esegue elaborazioni su di essi e li restituisce elaborati ai livelli di presentazione e data access;

- **livello data access**: per l'accesso alle basi dati, per esempio basi dati permanenti/persistenti come database relazionali, ma anche servizi di accesso a dati dinamici.

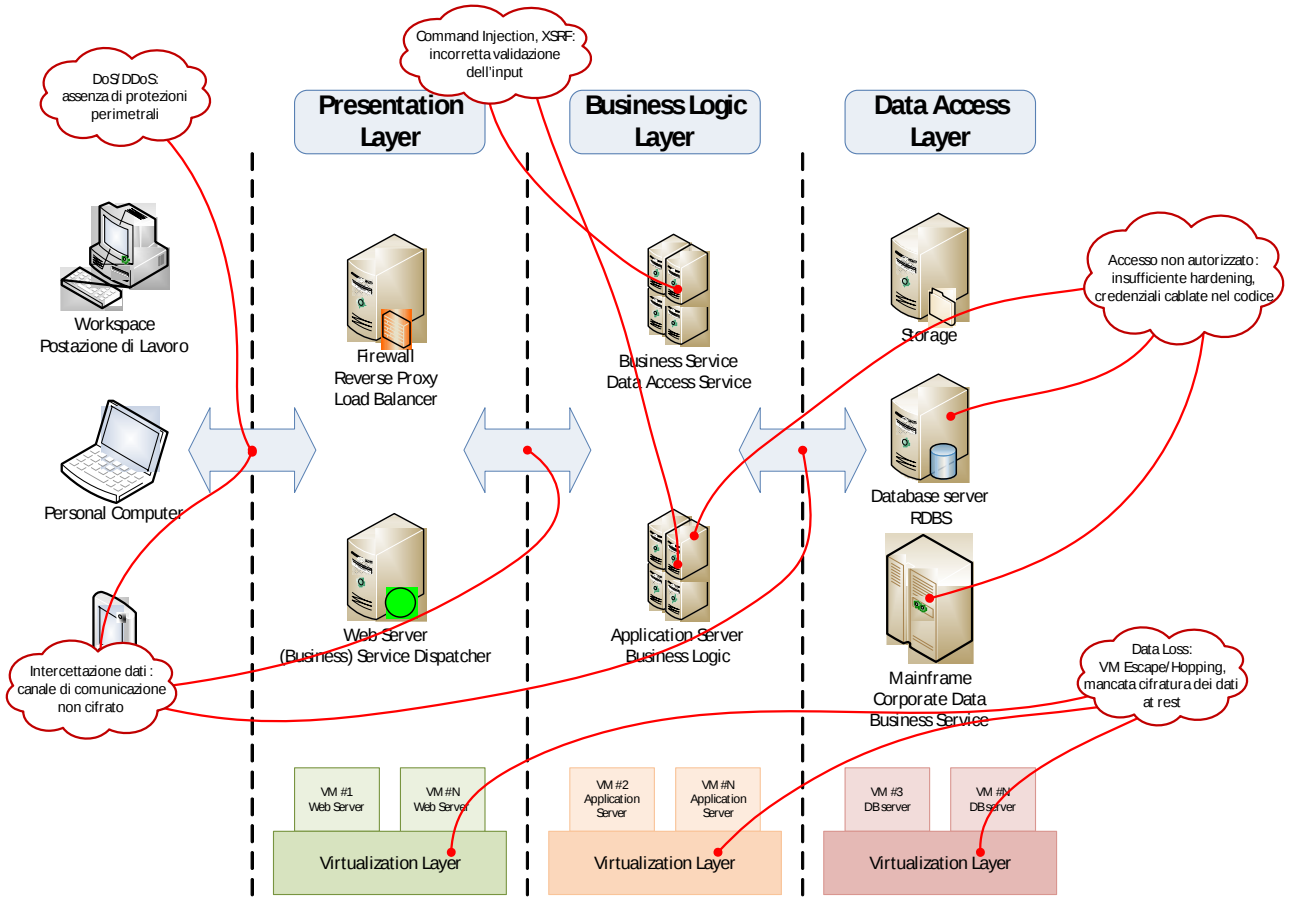
Gli attuali strumenti di sviluppo ed i componenti middleware sono generalmente modellati secondo questo schema architetturale. Le applicazioni devono, quantomeno a livello logico, seguire questo approccio di separazione dei livelli per garantire compartimentazione, separazione di privilegi, e modularità del software.

L'architettura logica di un'applicazione distribuita può essere quindi schematizzata come segue.



Contestualmente a questa suddivisione in diversi livelli, devono essere considerate le tecnologie di virtualizzazione, che da un lato consentono la razionalizzazione e l'ottimizzazione delle risorse hardware ma dall'altro introducono nuove minacce che devono esse tenute in considerazione durante tutto il ciclo di vita del software.

A titolo di esempio, sono illustrate alcune minacce tipiche dei tre livelli, con alcune problematiche che potrebbero subentrare a seguito dell'adozione di meccanismi di virtualizzazione.



3.3 Principi generali di sicurezza

In linea generale le applicazioni devono seguire i principi su cui si fonda la sicurezza di un qualunque prodotto software. Di seguito sono enucleati i principi fondamentali con relativa spiegazione ed esempio.

Defense in Depth / Difesa in profondità

Le applicazioni non devono utilizzare soltanto una protezione per una vulnerabilità o minaccia, ma valutare l'uso di più livelli di protezione. Inoltre, non bisogna mai essere confidenti sui livelli precedenti per la sicurezza del livello successivo, ed è necessario fare in modo che i livelli siano il più possibile indipendenti per segregare le informazioni ed attribuire le corrette responsabilità.

Un classico esempio è quello dei flussi informativi dal webserver al database. Supponiamo di avere un servizio esposto su Internet, il primo strato di difesa è costituito dal firewall che

consente di esporre soltanto la porta HTTP/HTTPS del servizio web. Il successivo strato di difesa è costituito dal webserver stesso che avrà delle opportune regole URL filtering/rewriting. I flussi che verranno fatti passare dal webserver, saranno a loro volta analizzati/validati/filtrati dall'application server, che valuterà i parametri della richiesta, prima di inoltrarli al database. Infine, l'ultimo blocco di difesa è costituito da un eventuale DAM (Database Activity Monitoring) attestato sul database, che deciderà se la query sta per accedere a tabelle normalmente non interrogate dall'application server.

Least Privilege & Need to Know

Il primo principio (Least Privilege) richiede che ogni attore abbia soltanto i privilegi minimi possibili per assolvere il suo lavoro; questo vale sia per gli utenti (cliente finale, amministratore, ecc …) che per componenti software (processi batch, ecc …).

Il secondo principio (Need to Know) richiede che ogni attore debba poter vedere il minor numero possibile di dati per assolvere il suo compito; questo vale sia per gli utenti che per componenti applicative che accedono ai dati.

L'unione dei due principi consente di arrivare ad una soluzione ottimale nella quale ogni attore ha soltanto tutti i dati e permessi necessari a compiere il proprio lavoro e nulla di più.

La corretta applicazione della combinazione di questi due principi dovrebbe evitare situazioni quali:

- l'applicazione è stata sviluppata partenndo dal presupposto che girerà con i massimi privilegi applicativi;
- l'applicazione accede a tabelle fuori dal suo schema e che dovrebbero essere visibili solo all'amministratore del database;
- utilizzo di un unico profilo per l'accesso a differenti schemi di database od a differenti database;
- accesso a qualsiasi file in lettura/scrittura/esecuzione sul filesystem.

Always Re-use

Evitare di reinventare da zero i moduli più critici (ad esempio per autenticazione ed autorizzazione) ma quando possibile riutilizzare sempre elementi basilari utili a produrre codice sicuro già sviluppati ed integrabili nel progetto, sotto forma di API del sistema operativo, o della specifica piattaforma (ad esempio: librerie Java, librerie .NET, ecc …).

Un esempio calzante risiede nei moduli di cifratura dei dati. Utilizzare una libreria standard Java o .NET per cifrare i dati (ad esempio con 3DES), consente di poterla riutilizzare senza dover riscrivere da zero codice che potrebbe essere soggetto a bug, e nel caso in cui non sia stata effettuata una efficace fase di code review a problematiche di sicurezza.

Avoid security by obscurity / Open design

È necessario evitare di progettare ed implementare applicazioni pensando che scrivere codice incomprensibile od utilizzare pattern difficilmente comprensibili consenta di innalzare, o addirittura raggiungere, i livelli di sicurezza desiderati.Bisogna pertanto prediligere sistemi e software sviluppati secondo modelli di sviluppo chiari e condivisi, algoritmi aperti e strumenti noti.

Come per il punto precedente, un esempio molto esplicativo riguarda la cifratura, dove il principio alla sua base è quello di mantenere segreta la chiave utilizzata per cifrare un determinato dato e non l'algoritmo. Infatti, l'utilizzo di un algoritmo standard (ad esempio 3DES) in combinazione con l'applicazione delle best practice associate, creerà un meccanismo di cifratura sicuramento più robusto di quello che si potrebbe ottenere sviluppandone un proprio od utilizzandone uno proprietario, che potrebbe risultare potenzialmente debole.

KISS (Keep It Simple Stupid)

Questo principio segue quello di Open Design, ovvero rimarca il concetto di utilizzare sempre soluzioni semplici sia nella progettazione del software (ad esempio pattern noti), che nella scrittura di codice evitando quindi inutili complicazioni. Pertanto è necessario prediligere codice/software/architetture comprensibili sia agli addetti ai lavori che ad altri attori, senza ricorrere a complesse astrazioni per la loro spiegazione ma utilizzando semplici metafore e/o analogie.

Un classico esempio è rappresentato dall'operatore ternario presente nella maggior parte dei linguaggi di programmazione (<condizione> ? <caso vero> : <caso falso>). Questo è normalmente molto utilizzato dai programmatori più esperti, in quanto velocizza la scrittura di codice, ma al tempo stesso rende la lettura del codice più complessa, e forse incompressibile ai non addetti ai lavori, rispetto ai classici blocchi if then ed else.

Detect intrusion / compromise recording

Nella fase di progettazione e realizzazione del software è necessario non sottovalutare/ignorare le esigenze di tracciamento. In linea generale è sempre utile poter capire cosa è successo sia a seguito di fault del software stesso sia in termini di "chi ha fatto cosa".

Pertanto è necessario progettare il software in modo tale che siano sempre possibile individuare comportamenti imprevisti e/o illeciti utilizzando:

- log: con un sufficiente dettaglio che consenta la ricostruzione degli eventi, ma che non contenga informazioni puntuali sui dati processati (ad esempio contiene riferimenti temporali ed IP ma non eventuali dati sensibili gestiti dall'applicativo);
- sistemi di monitoraggio: l'applicazione dovrebbe prevedere opportune interfacce per poter comunicare con i SIEM.

Positive Security Model / White list

Il principio può essere riassunto con la frase: definire cosa è permesso e rigettare tutto il

resto. Più in generale, in qualunque caso sia possibile, evitare controlli che blocchino funzionalità per il caso negativo ma prediligere controlli che consentano l'erogazione di funzionalità solo nel caso positivo.

Uno degli esempi più comprensibili riguarda il mondo dei firewall, dove è possibile usare un approccio non corretto di tipo black list, ad esempio bloccando solo un insieme di IP che rientrano nella black list stessa. Invece, andrebbe utilizzato l'approccio white list che prevede di accettare connessioni solo da IP accreditati, ovvero in white list.

Un altro esempio, più incentrato al mondo dello sviluppo, soprattutto di web application, è quello di accettare soltanto un insieme di caratteri permessi (ad esempio numeri [0-9] e lettere [a-Z]), e non quello di prevedere meccanismi che scartino un insieme di caratteri non permessi (ad esempio scartare solo l'apice ['']), in quanto altre combinazioni di caratteri potrebbero compromettere l'applicazione.

Fail-Safe / Fail-Secure / Fail Securely

In caso di fault od errori a runtime, l'applicazione deve comunque andare in uno stato nel quale non siano compromesse le risorse gestite. Ovvero, è necessario gestire correttamente le eccezioni di qualunque tipo. Contestualmente devono essere gestite anche le informazioni/dati ad esse associate in quanto possono fornire preziosissime informazioni ad un attaccante (Information Leak / Disclosure). Infine, un fallimento deve comunque far rientrare il comportamento dell'applicazione in un binario previsto/progettato.

In questo caso un esempio è costituito dalla classica pagina di errore (sia per applicazioni J2EE che .NET), generata dall'application server a causa di un inadeguato hardening, e contenete tutto lo stack della problematica. In questo caso l'approccio ideale sarebbe quello suggerito dal principio di Compromise Recording, ovvero che in caso di falut dell'applicazione, l'application server riporti l'errore solo nei log o gli inoltri ad un sistema di monitoraggio.

Un altro approccio, spesso usato nel caso di applicazioni web, è quello di sfruttare i costrutti try/catch/finally, presenti in tutti i linguaggi di programmazione, e mettere in questi le linee di codice che potrebbero generare errori. Così è possibile far in modo che in caso di errore l'applicazione reindirizzi l'utente verso la pagina principale, richiedendo di ripetere l'operazione.

Don't trust infrastructure

Le applicazioni non devono riporre eccessiva sicurezza nell'infrastruttura su cui sono istallate, e soprattutto non devono basare la protezione dei dati su di essa. Questa tematica diventa ancora più critica con la virtualizzazione, in quanto i dati risiedono su un'infrastruttura condivisa e soprattutto le virtual machine possono essere spostate da un server fisico ad un altro.

Inoltre, è estremamente probabile che l'infrastruttura cambi ad un ritmo insostenibile per valutarne gli impatti e creare patch.

In questi casi diventa critico progettare l'applicazione affinché gli elementi più a rischio del

software siano protetti indipendentemente dalle altre misure di sicurezza implementate.

A tal proposito, un esempio riguarda i dati cosiddetti at rest (ovvero persistenti) si pensi ad una virtual machine che mantiene i propri dati residenti sul filesystem virtualizzato e che viene migrata improvvisamente per ragioni di prestazioni hardware. In questo caso se i dati su disco non sono stati cifrati potrebbero rimanere sul disco fisico del server che eroga la virtualizzazione, ed essere acceduti da un'altra virtual machine che viene eseguita sulla stessa allocazione, od in caso di incorretto smaltimento dell'hardware potrebbe rimanere in uno stato intellegibile.

Don't trust service

Il termine servizi fa riferimento a qualsiasi interfacciamento/interazione con un sistema esterno con il quale l'applicativo coopera. Questi possono essere potenziali vettori di attacchi ed inoltre i sistemi esterni potrebbero non essere soggetti agli stessi regolamenti dell'Ente e pertanto avere policy di sicurezza più lasche.

In questo caso non bisogna riporre eccessiva fiducia nei sistemi gestiti esternamente, anche in caso di fornitori enterprise, ma è necessario definire delle policy di sicurezza sempre sulla base dei dati trattati e delle informazioni fornite, avendo cura di progettare l'applicazione in modo tale da non processare le informazioni senza prima valutarne la bontà o richiamare funzionalità esterne senza prevedere opportuni meccanismi di controllo sull'esito dell'elaborazione esterna.

Un esempio riguarda i servizi SOA e relative logiche di business collegate, nelle quali non deve essere dato per scontato che i valori prodotti da servizi esterni siano affidabili, ma è necessario comunque analizzare i risultati prodotti sia per coerenza delle informazioni sia per evitare che i dati prodotti possano contenere codice malevole (ad esempio virus, ma anche command injection).

Establish Secure Default

In generale è buona norma che tutte le impostazioni di default di applicazioni/middleware/database/sistemi operativi siano quelle che garantiscono il maggior livello di sicurezza o quello desiderato, consentendo comunque ai vari attori la modifica di alcuni parametri per attività di tuning o configurazioni ad hoc.

In generale il principio si propone di evitare situazioni quali: non dare, di default, elevati privilegi ad un utente sperando che sia lui ad abbassarli se necessario.

Altri esempi che riguardano le web application possono essere: attivare sempre il modulo per accettare solo password complesse dagli utenti; utilizzare solo protocolli di comunicazioni cifrati quali HTTPS; disattivare le console di configurazione remota degli application server od esporle solo su interfacce locali; ecc …

The Weak Point / Il punto debole della catena

La sicurezza di un'architettura o di un generico insieme di componenti è pari alla sicurezza dell'anello più debole. Pertanto è importante individuare e gestire correttamente la

sicurezza dell'elemento meno sicuro della catena, in quanto la sicurezza del sistema è la sicurezza della componente meno sicura.

Vista la natura stessa del concetto di weak point, non è possibile fornire un esempio esplicativo. Quello che è possibile fornire è un buon esempio della loro gestione, che normalmente consiste nell'introduzione di procedure standard per la gestione degli asset in modo da favorirne l'individuazione e la successiva gestione:

- processo per il censimento degli asset ed aggiornamento del configuration management database: sistemi, applicazioni, software generico, hardware, ecc …
- procedure per updating/patching/hardening;
- processi per la raccolta e gestione dei log con sistemi allarmistica;
- introduzione di IDS, IPS, ecc …

4. Design e sviluppo dell'applicazione

I requisiti di sicurezza sono un insieme di qualità richieste al software ma anche a tutte le componenti a supporto (servizi centralizzati, strutture a supporto, processi, ecc …).

Nei paragrafi seguenti sono riportate, suddivisi per macro categorie e con indicazione delle minacce che vanno a coprire, tutte le contromisure di sicurezza che un'applicazione deve adottare al fine di essere in linea con i requisiti di sicurezza richiesti dall'Ente.

Le indicazioni riportate nei paragrafi successivi saranno integrate da appositi Allegati tecnici, redatti successivamente all'adozione di questo disciplinare. Tali Allegati tecnici, uno per ogni filiera applicativa supportata dall'Ente, avranno lo scopo di descrivere in maniera più approfondita e specifica le modalità con cui le applicazioni delle diverse filiere potranno realizzare le contromisure alle minacce indicate nei paragrafi successivi.

4.1 Analisi dei requisiti e design

Minacce

Accesso non autorizzato alle risorse, esecuzione di operazioni non consentite, mancata disponibilità dei servizi e dei dati, mancato rispetto degli obblighi previsti dalla normativa vigente.

Contromisure

A. Identificare, in fase di raccolta dei requisiti, la natura ed il valore dei dati e delle informazioni che saranno trattate dall'applicazione. Individuare in particolare, facendo riferimento alla normativa vigente, se si trattino le seguenti categorie di dati personali:
   a. dati personali non sensibili e giudiziari;
   b. dati sensibili;
   c. dati giudiziari.

B. Eseguire, successivamente all'individuazione della natura e del valore dei dati e delle informazioni, l'analisi dei rischi incombenti su di esse in relazione alle minacce ed alle contromisure di sicurezza disponibili. L'analisi dei rischi è elemento fondamentale per la scelta delle misure di sicurezza appropriate. Tale scelta deve essere fatta in funzione del valore delle risorse da proteggere e dei danni che una eventuale compromissione della sicurezza comporterebbe. L'Appendice A contiene un esempio sintetico di analisi dei rischi effettuata per un'applicazione web.

X. Considerare eventuali vincoli infrastrutturali e tecnologici che possono influire sul comportamento dell'applicazione e comprometterne la sicurezza (per es. restrizione di porte o protocolli, piattaforme tecnologiche di sviluppo, ecc …).

Δ. Censire con precisione, in fase di design, le porte ed i protocolli utilizzati dall'applicazione, in modo da predisporre un ambiente di produzione che limiti all'indispensabile la superficie di attacco dell'applicazione stessa.

E. Progettare l'applicazione prevedendo l'implementazione dei seguenti meccanismi di sicurezza:

a. autenticazione;
b. autorizzazione;
c. gestione delle sessioni utente;
d. validazione dei dati;
e. gestione degli errori;
f. tracciamento;
g. monitoraggio;
h. cifratura dei dati at rest ed in motion;
i. disponibilità dei dati;
j. documentazione di progetto;
k. codice applicativo;
l. applicazioni per dispositivi mobili.

Φ. Utilizzare la lista di controllo "Analisi dei requisiti e design" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.2 Autenticazione

Minacce

Accesso non autorizzato alle risorse, spoofing, individuazione/furto delle credenziali/password, privilege escalation.

Contromisure

A. Stabilire in fase di design dove ed in che modo sia necessario garantire l'autenticazione (di utenti od entità). Tale scelta deve essere fatta in funzione del valore delle risorse da proteggere e delle policy dell'Ente in materia di autenticazione.

B. Stabilire il meccanismo di autenticazione e la scelta delle credenziali da utilizzare, in funzione del valore delle risorse da proteggere. Le credenziali di autenticazione possono consistere in un codice per l'identificazione dell'utente associato a una parola chiave riservata conosciuta solamente dal medesimo (per es. userid/password) oppure in un dispositivo di autenticazione in possesso ed uso esclusivo dell'utente (per es. smartcard o token hardware), eventualmente associato ad un codice identificativo o parola chiave, oppure in una caratteristica biometrica dell'utente (per es. impronta digitale), eventualmente associata a un codice identificativo o parola chiave.

C. Utilizzare, dove possibile, i meccanismi di autenticazione centralizzati in uso nell'Ente, in modo che l'autenticazione non sia parte del codice applicativo, ma sia basata su meccanismi dedicati (per es. Access Manager, LDAP, Active Directory, ecc …). In questo modo si realizza una duplice ottimizzazione: da un lato l'utente non è costretto a ricordare una nuova userid/password (all'aumentare del numero di password utilizzate aumenta infatti la probabilità che esse siano dimenticate, annotate per iscritto, scelte con meno cura, utilizzate su più sistemi diversi), dall'altro non si utilizza un database dedicato per la memorizzazione e gestione credenziali a livello applicativo.

D. Prevedere, in caso di autenticazione basata su password, i seguenti meccanismi di sicurezza obbligatori per legge:

   a. scadenza della password: prevedere meccanismi di controllo della scadenza della validità della password, che deve essere inferiore ai 180 giorni (90 giorni nel caso di applicazioni relative al trattamento di dati sensibili e/o giudiziari);

   b. lunghezza della password: prevedere meccanismi di controllo sulla lunghezza della password, che deve essere di almeno 8 caratteri;

   c. modifica al primo login: prevedere meccanismi che consentano la modifica della propria password da parte dell'utente al primo accesso al sistema.

E. Prevedere ulteriori meccanismi di sicurezza per i sistemi basati su password dove valutato necessario dall'analisi dei rischi effettuata. In particolare:

   a. meccanismi di lockout: prevedere la disabilitazione di un account dopo un intervallo finito di tentativi di accesso non riusciti (per contrastare gli attacchi alle password di tipo bruteforce). Prevedere meccanismi di difesa da attacchi di tipo denial-of-service causati dal blocco volontario di account legittimi (per es. bloccando un account per poi riabilitarlo trascorsi 10' dal blocco, oppure prevedendo un delay di 5" a seguito di un'autenticazione errata);

   b. meccanismi di reset della password: prevedere meccanismi che consentano all'utente di modificare la propria password senza l'intervento degli amministratori di sistema;

   c. meccanismi di cifratura delle password: prevedere meccanismi di conservazione protetta delle password, che non devono mai essere conservate o trasmesse in chiaro (per es. utilizzo di hash per la memorizzazione delle password, utilizzo di protocolli di comunicazione cifrati come HTTPS, ecc ...);

   d. meccanismi di controllo della robustezza delle password: prevedere meccanismi di controllo che consentano esclusivamente l'utilizzo di password corrispondenti a determinati criteri di complessità (per es. password con almeno un valore letterale maiuscolo, con almeno un valore numerico, con almeno un carattere simbolico, ecc …).

F. Prevedere meccanismi di disattivazione delle credenziali non utilizzate da almeno 180 giorni.

G. Ridurre al minimo le informazioni fornite in caso di errore di autenticazione, nello specifico nel caso di:

   a. credenziali riportare soltanto frasi quali “combinazione username/password errata”, e non errori che possono consentire di enumerare gli utenti come “password errata” o “utente non valido”;

   b. dispositivi: riportare soltanto frasi quali “certificato non valido” e non informazioni di debugging remoto quali “CA non riconosciuta”;

   Informazioni superflue sono utili ad un attaccante per comprendere i meccanismi di autenticazione del sistema ed i metodi per aggirarli.

H. Non riutilizzare i codici di identificazione già impiegati assegnandoli ad altri utenti (neanche in tempi diversi).

I. Per l’autenticazione machine-to-machine e/o application-to-application prediligere l’uso di tecniche di autenticazione forte mediante l’uso di certificati digitali e chiavi asimmetriche.

J. Utilizzare la lista di controllo "Autenticazione" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.3 Autorizzazione

Minacce

Accesso non autorizzato ai dati, modifica non consentita di dati, esecuzione di operazioni non consentite.

Contromisure

A. Implementare meccanismi di separazione dei privilegi per garantire l’utilizzo delle risorse in funzione di differenti profilazioni degli utenti.

B. Utilizzare il principio del "minimo privilegio" nell'attribuzione dei permessi, ovvero abilitare l’accesso alle sole risorse indispensabili e negarlo a tutte le restanti.

C. Limitare al minimo, ed evitare dove possibile, l’accesso alle risorse di sistema (file, cartelle, registry, log, ecc …).

D. Non consentire al login applicativo l'accesso diretto in scrittura alle tabelle dei database di backend. Utilizzare credenziali di autenticazione ai database con i privilegi minimi indispensabili.

E. Stabilire, in caso di applicazioni web, dove sia necessario applicare la separazione dei privilegi tra aree web ad accesso pubblico ed aree web ad accesso riservato.

F. Quando possibile utilizzare, per il controllo degli accessi degli utenti, i sistemi di sicurezza del framework utilizzato: “Java security policy” per J2EE e/o “Security-Transparent Code, Level 2” in ambiente .NET;

G. Utilizzare la lista di controllo "Autorizzazione" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.4 Gestione delle sessioni utente

Minacce

Spoofing, session hijacking.

Contromisure

A. Prevedere meccanismi di protezione delle credenziali utilizzate per il riconoscimento dell’utente dopo il login.

B. Limitare la durata delle sessioni ad un periodo di tempo definito in funzione delle caratteristiche funzionali dell'applicazione e prevedere il blocco della sessione allo scadere di tale periodo.

C. Non trasferire mai in chiaro gli identificatori di sessione (per es. token, stringhe di query, ecc …). Nelle applicazioni web proteggere i cookie di autenticazione di sessione tramite l’utilizzo del protocollo TLS o cifrandone il contenuto.

D. Non utilizzare algoritmi custom per la generazione degli identificativi di sessione (ad esempio per token), ma affidarsi al framework od application server sul quale è stata sviluppata l’applicazione.

E. Implementare meccanismi di logout che permettano all'utente di forzare la chiusura di una sessione.

F. Nel caso di sessioni web autenticate, utilizzare (impostando a true) i seguenti attributi dei cookie di sessione:

   a. Secure (solo nel caso si utilizzi https): in modo che le informazioni vengano inviate dal browser soltanto su canale cifrato;

   b. HttpOnly: in modo da non consentire ad Applet o JavaScript di accedere ai parametri del cookie.

G. Utilizzare la lista di controllo "Gestione delle sessioni utente" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.5 Validazione dei dati

Minacce

Stringhe “nocive” inserite in query, form, cookie e header HTTP. Esecuzione di comandi, cross-site scripting (XSS), Cross-site request forgery (CSRF/XSRF), SQL injection, LDAP injection, buffer overflow, denial-of-service.

Contromisure

A. Validare sia l’input sia l’output per controllare che siano rispondenti a quanto l’applicazione si aspetta in termini di:
   a. formato dei dati;
   b. sintassi;
   c. dimensioni.

B. Considerare sempre inattendibili, e quindi da validare, tutti i dati in input e output.

C. Effettuare tutte le validazioni dei dati lato server. Dove per comodità vengano implementate validazione lato client, esse devono essere accessorie a quelle effettuate lato server. Le strategie di validazione possibili sono:
   a. accettare solo dati riconosciuti validi;
   b. rigettare i dati riconosciuti come non validi;
   c. modificare i dati non validi per renderli validi.

   Preferire, dove tecnicamente realizzabile, la soluzione (a) (i dati riconosciuti validi rimangono costanti nel tempo, mentre i dati non validi possono cambiare nel tempo con l’evolversi delle tecniche di attacco). Impiegare la soluzione (c) solo come misura aggiuntiva ad una delle precedenti. La soluzione ideale comprende tutte e tre le strategie di validazione.

D. Adottare un sistema di firma per certificare i dati che provengono o vengono inviati ad altre applicazioni ed accettare dati esclusivamente da sistemi riconosciuti e fidati.

E. Per la prevenzione di SQL injection prediligere, nell’ordine, l’utilizzo di:
   a. Prepared Statement / Parameterized Query;
   b. Stored Procedure: non utilizzare query dinamiche all’interno del corpo delle stored procedure;
   c. escaping dei caratteri: prediligere l’utilizzo di librerie consolidate (ad es. le ESAPI di OWASP) che supportino i char set del database su cui si basa l’applicazione;

F. Per la prevenzione di LDAP injection prevedere sempre controlli sui caratteri: asterisco, uguale, parentesi tonda aperta e parentesi tonda chiusa;

G. Per la prevenzione da attacchi di tipo XSS, far in modo di:
   a. non inserire mai il contenuto di parametri HTTP all’interno di codice JavaScript preesistente, CSS o tag HTML non chiusi;
   b. rimuovere o convertire caratteri speciali come parentesi uncinate/angolari/acute;

c. utilizzare funzioni apposite per l'inserimento dinamico di testo all'interno della pagina, che codificano in modo automatico i caratteri speciali;

Per ulteriori dettagli fare riferimento ad OWASP, nell'apposita sezione "XSS (Cross Site Scripting) Prevention Cheat Sheet".

H. Per la prevenzione di attacchi di tipo XSRF/CSRF inserire degli appositi token nelle richieste "critiche" al server, avendo cura di:

a. non inserire i token in link o qualsiasi altro riferimento che possa generare richieste di tipo GET;

b. quando possibile utilizzare sistemi di prevenzione nativi del framework o application server sul quale è stata sviluppata l'applicazione (ad esempio in .NET abilitare il Viewstate);

c. utilizzare tecniche come "Double Submit Cookie" che prevedono di inviare parametri, nello specifico il session ID, tramite differenti canali (ad esempio sia tramite cookie che come parametro nascosto nelle FORM).

Per ulteriori dettagli fare riferimento ad OWASP, nell'apposita sezione "Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet".

I. Utilizzare la lista di controllo "Validazione dei dati" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.6 Gestione degli errori

Minacce

Information Leakage, Information Disclosure, Accesso non autorizzato ai dati, modifica non consentita di dati, esecuzione di operazioni non consentite.

Contromisure

A. Le applicazioni devono essere progettate prevedendo che in caso di un'anomalia o di un errore software, il flusso operativo vada sempre a finire in uno stato "sicuro" (fail safe).

B. Le funzionalità di logging e debugging devono essere configurabili per riportare messaggi dettagliati solo negli ambienti di verifica e di sviluppo, mentre negli ambienti di esercizio devono essere configurate per riportare solo un dettaglio minimo per i messaggi di errore.

C. Sviluppare, quando possibile, l'applicazione utilizzando linguaggi di programmazione che consentono la gestione strutturata degli errori (costrutti come try/catch/finally in Java e C#), garantendo maggiore controllo delle possibili anomalie del software e mantenendo il codice leggibile.

D. Utilizzare la lista di controllo "Gestione degli errori" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.7 Tracciamento

Minacce

Mancato rilevamento di intrusioni, impossibilità di dimostrare un'eventuale azione illecita compiuta da un utente, difficoltà nel diagnosticare malfunzionamenti ed anomalie di funzionamento dell'applicazione.

Contromisure

A. Definire in fase di design quali sono gli eventi chiave per la sicurezza dell'applicazione da rilevare tramite tracciamento/logging.

B. Registrare nei log, dove tecnicamente possibile, i seguenti eventi applicativi:
   a. autenticazione applicativa (login e logout, riusciti e non);
   b. accesso ai dati (lettura e scrittura);
   c. modifica di funzioni amministrative (per es. la disabilitazione delle funzioni di logging, la gestione dei permessi, ecc …).

C. Prevedere meccanismi di controllo e modifica del livello di granularità dei dati rilevabili.

D. Prevedere la possibilità di registrare, all'interno di un voce di log, le seguenti informazioni:
   a. data ed ora dell'evento;
   b. luogo dell'evento (ad esempio: macchina/hostname, indirizzo IP, ecc …);
   c. identificativo dell'entità che ha generato l'evento (ad esempio: utente, servizio, processo, ecc …);
   d. descrizione dell'evento.

E. All'interno dei log non devono essere presenti informazioni critiche per l'Ente, ma soltanto quelle informazioni che consentono la ricostruzione di eventi o problematiche.

F. Prevedere, se possibile, meccanismi di interconnessione con il sistema regionale di gestione dei log (SIEM), al fine di garantire la loro conservazione in modalità sicura. Se è possibile, al fine di facilitare l'interconnessione con il SIEM, utilizzare per la produzione dei log il formato CEF (Common Event Format).

G. Nel caso non fosse possibile l'interconnessione con il SIEM, prevedere la conservazione dei log in file in cui sia possibile effettuare la scrittura incrementale o su supporti non riscrivibili; prevedere inoltre meccanismi di backup dei log secondo le procedure di backup centralizzato previste dall'Ente.

H. Prevedere meccanismi di sovrascrittura dei log esistenti ad intervalli regolari. La frequenza di rotazione dei log è dipendente dalla frequenza di generazione degli eventi dell'applicazione. Devono in ogni caso essere rispettate le policy dell'Ente in materia di trattamento dei log.

I. Prevedere meccanismi di controllo degli accessi ai log tramite autenticazione ed autorizzazione. L'accesso ai log deve poter essere eseguito solo da utenti privilegiati (per es. membri del gruppo *Administrators* nei sistemi Windows based o *root* nei sistemi UNIX) e comunque, salvo esigenze particolari e documentate, secondo le policy dell'Ente in materia di trattamento dei log.

J. Prevedere meccanismi di verifica periodica del corretto funzionamento dei sistemi di logging.

K. Utilizzare la lista di controllo "Tracciamento" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.8 Monitoraggio

Minacce

Mancato rilevamento di intrusioni, difficoltà nel diagnosticare malfunzionamenti ed anomalie di funzionamento dell'applicazione.

Contromisure

A. Definire in fase di design cosa deve essere monitorato dell'applicazione.Si consiglia di monitorare almeno le seguenti informazioni:
   a. stato dell'applicazione (ad esempio: in esecuzione, non in esecuzione);
   b. utilizzo delle risorse http(ad: esempio: response time, numero di connessioni)ed altri dati che consentano un'efficiente/efficace fase di troubleshooting.

B. Prevedere, se possibile, meccanismi di interconnessione con la piattaforma di monitoraggio centralizzato dell'Ente.

C. Utilizzare la lista di controllo "Monitoraggio" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.9 Cifratura dei dati at rest ed in motion

Minacce

Accesso non autorizzato ai dati, accesso a credenziali utenti, attacchi man in the middle.

Contromisure

A. Considerare l'utilizzo di meccanismi di cifratura dei dati (sia per la conservazione sia per la trasmissione) in funzione del valore delle risorse da proteggere e delle minacce di sicurezza, con particolare attenzione ai casi in cui l'applicazione sia destinata al trattamento di dati sensibili e/o giudiziari e/o critici per l'Ente.

B. I dati sensibili/giudiziari/critici devono rimanere in chiaro solo per il tempo necessario alla loro elaborazione, e quando possibile fare in modo che le chiavi di cifratura

risiedano su sistemi differenti da quelli in cui vengo mantenuti i dati at rest (dati salvati in qualche dispositivo fisico/logico).

C. Utilizzare meccanismi di firma digitale nei casi in cui sia necessario garantire la non ripudiabilità delle transazioni. Prevedere, in tal caso, chiavi di firma distinte dalle chiavi di cifratura.

D. Non conservare password di autenticazione e chiavi private o di cifratura in chiaro. Cifrare sempre le password e le chiavi di cifratura, proteggere le chiavi private con passphrase.

E. In caso di dati in motion (trasferimento dati fra dispositivi fisici/logici) valutare se debbano essere inviati cifrati, non trasmettere mai in chiaro credenziali di autenticazione o dati sensibili/giudiziari/critici:

   a. per questo tipo di dati utilizzare almeno flussi cifrati su TLS/SSLv3;

   b. per i flussi dati diretti verso sistemi all'interno dell'Ente utilizzare una CA (Certification Autority) riconosciuta all'interno dell'Ente stesso;

   c. per i flussi diretti verso l'esterno, l'applicazione deve utilizzare un certificato riconosciuto e firmato da una CA (Certification Autority) attendibile;

   d. in caso di mutua autenticazione, o di connessione verso servizi esterni, verificare sempre tutta la catena dei certificati.

F. Definire, in fase di design dell'applicazione, gli algoritmi da utilizzare e la lunghezza delle chiavi di cifratura in funzione dell'importanza delle risorse da proteggere e del progresso tecnico nel campo della sicurezza informatica e della crittoanalisi.

G. Utilizzare algoritmi di cifratura standard e robusti, con chiavi di cifratura aventi lunghezza adeguata. A titolo di esempio:

   a. algoritmi di cifratura simmetrici: prediligere l'utilizzo di Advanced Encryption Standard (AES) o Triple DES (3DES) con chiavi di lunghezza minima pari a 128 bit, utilizzare altri algoritmi e lunghezze di chiavi inferiori solo in caso di compatibilità con sistemi legacy;

   b. algoritmi di cifratura asimmetrici: prediligere l'utilizzo di Rivest-Shamir-Adleman (RSA) o Digital Signature Algorithm (DSA) con chiavi a 2.048 bit o oltre utilizzare altri algoritmi o chiavo solo in caso di compatibilità con sistemi legacy;

   c. funzioni di hash dei messaggi: prediligere SHA-224 o superiori, ed utilizzare MD5 o SHA-128 solo in caso di compatibilità con sistemi legacy.

H. Prevedere adeguati meccanismi di gestione sicura delle chiavi di cifratura. Per esempio:

   a. meccanismi di distribuzione delle chiavi;

   b. meccanismi di conservazione delle chiavi;

   c. meccanismi di riciclo periodico delle chiavi;

   d. meccanismi di revoca delle chiavi;

   e. meccanismi di recovery delle chiavi;

   f. meccanismi di distruzione delle chiavi.

I. Nel caso in cui le applicazioni risiedano su sistemi virtualizzati, valutare la possibilità di cifrare sempre e comunque lo spazio disco su cui risiedono i dati, in modo tale da evitare che, in caso di riassegnazione dello storage fisico od inadeguato smaltimento dei dispositivi, i dati possano essere facilmente recuperati.

J. Utilizzare la lista di controllo "Cifratura dei dati" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.10 Disponibilità dei dati

Minacce

Mancata disponibilità dei dati, interruzione della continuità operativa.

Contromisure

A. Definire, in fase di design, il valore di impatto operativo sul quale saranno basate le politiche di disaster recovery e continuità operativa dei sistemi che contribuiscono all'erogazione dell'applicazione, come da documentazione da presentare prima della presa in carico di una nuova applicazione, secondo quanto sarà previsto dalle "Linee Guida per la governance del sistema informatico regionale".Definire, in fase di design, la sequenza esatta di avvio e arresto dei servizi necessari all'applicazione, nonché l'interazione con altre applicazioni, come da documentazione da presentare prima della presa in carico di una nuova applicazione, secondo quanto sarà previsto dalle "Linee Guida per la governance del sistema informatico regionale".

B. Prevedere meccanismi di backup attraverso punti di sincronizzazione aperti e compatibili con le policy di backup dell'Ente.

C. Prevedere, nel caso l'applicazione sia utilizzata per il trattamento di dati personali, meccanismi di backup dei dati con frequenza almeno settimanale.

D. Prevedere, nel caso l'applicazione sia utilizzata per il trattamento di dati sensibili e/o giudiziari, meccanismi di ripristino dei dati in tempi non superiori ai sette giorni.

E. Utilizzare la lista di controllo "Disponibilità dei dati" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.11 Documentazione di progetto

Minacce

Lock-in soprattutto rispetto ad interoperabilità e portabilità.

Contromisure

A. Redigere una documentazione di progetto utilizzando standard internazionali (ed esempio: UML, ER, Use Cases, ecc …) in modo da consentire una facile analisi e comprensione dell'applicazione. All'interno della documentazione devono essere

previste apposite sezioni per dare evidenze delle scelte architetturali più importanti (ad esempio pattern architetturali), delle modalità di integrazioni con terze parti, ed eventuali altre informazioni utili a promuovere l'interoperabilità con altri sistemi e la portabilità dei dati.

B. Prevedere apposita documentazione che censisca, quanto più dettagliatamente possibile, i flussi informativi e le modalità di comunicazione con altri attori o sistemi (ad esempio il server accetta flussi HTTPS sulla porta 443, da qualsiasi host per richieste di autenticazione), come da documentazione da presentare prima della presa in carico di una nuova applicazione, secondo quanto previsto dalle "Linee Guida per la governance del sistema informatica regionale".

C. Prevedere apposita documentazione per la gestione degli errori applicativi e dell'applicativo, in modo da consentire l'attuazione delle procedure in caso di errori.

D. Utilizzare la lista di controllo "Documentazione di progetto" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

4.12 Codice applicativo

Minacce

Accesso non autorizzato a dati, Information Leakage, Information Disclosure.

Contromisure

A. In caso di codice sorgente che può essere visualizzato dall'utente finale, non inserire all'interno dei commenti del codice dati sensibili, critici, o che consentano di comprendere la logica dell'applicazione.

B. Non inserire/cablare all'interno del codice credenziali (username/password, certificati), chiavi di cifratura, o passphrase, soprattutto in caso di codice sorgente che può essere visualizzato dall'utente finale.

C. Prevedere delle verifiche, come attività di codereview, affinché il codice non contenga costrutti, API, librerie notoriamente affette da problematiche o vulnerabilità.

D. Quando possibile fare in modo che il codice (ad esempio per: JavaScript, Java, .NET) delle applicazioni sia adeguatamente offuscato quando rilasciato in ambienti di esercizio.

E. Utilizzare la lista di controllo "Codice applicativo" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

A.

4.13 Applicazioni per dispositivi mobili

Di seguito sono enucleati alcuni accorgimenti, aggiuntivi rispetto a quelli indicati nei paragrafi precedenti, da applicare nella fase di sviluppo di applicazioni per dispositivi mobili. L'elenco non è da considerarsi esaustivo, ma deve essere utilizzato come spunto per individuare e coprire quelle che sono le nuove minacce derivanti dall'uso di dispositivi mobili, sia per applicazioni distribuite al pubblico che usate all'interno dell'Ente (in particolare nei casi di utilizzo di dispositivi mobili di proprietà dei collaboratori dell'ente, pratica nota come BYOD – bring your own device).

Minacce

Accesso non autorizzato ai dati, session hijacking, Information Leakage, Information Disclosure, Wi-Fi Sniffing, attacchi man in the middle, furto o smarrimento del dispositivo, code injection, code tampering.

Contromisure

A. Evitare di utilizzare lo storage condiviso dei dispositivi per il salvataggio di credenziali, token di autenticazione, chiavi di cifratura, ma quando possibile utilizzare soltanto le aree di memoria dedicate all'applicazione stessa.

B. Non utilizzare le memorie rimovibili (ad esempio SD Card) per salvare, anche in maniera temporanea o cifrata, dati sensibili/giudiziari/critici, in modo da salvaguardarne la riservatezza anche in caso di furto o smarrimento del dispositivo.

C. Devono essere previsti appositi meccanismi per impedire l'utilizzo di applicazioni che trattano dati sensibili/giudiziari/critici su dispositivi manomessi ovvero sui quali sia possibile ottenere i massimi privilegi (ad esempio in caso di Jailbreak/root del dispositivo).

D. All'interno delle applicazioni non devono essere inseriti certificati per la verifica dell'identità del server, ma deve essere sempre e comunque verificata tutta la catena dei certificati (certificate trust chain) tramite certificate authority riconosciute a livello nazionale/interinazione, in modo da prevenire attacchi man in the middle (ad esempio fake/rouge Wi-Fi hotspot).

E. La validazione dei dati deve avvenire, oltre che lato server, anche sull'applicazione mobile sia per i flussi dati inviati al server che per quelli provenienti dal server, onde evitare la compromissione del dispositivo stesso.

F. Per rispettare il principio del minimo privilegio, minimizzare le richieste di permesso a quelle necessarie per il funzionamento dell'applicazione. Ogni richiesta di permesso deve essere motivata.

G. Non utilizzare il caricamento dinamico del codice dall'esterno dell'applicazione, per ridurre il rischio di code injection o code tampering e rendere sempre possibile la verifica del comportamento dell'applicazione.

H. Utilizzare la lista di controllo "Applicazioni mobili" (Appendice B.1) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

5. Test, deployment e gestione dell'applicazione

Alcune delle seguenti contromisure non sono direttamente a carico dello sviluppo ma dalle strutture di esercizio. Tali contromisure sono comunque importanti e pertanto devono essere attuate onde vanificare gli sforzi fatti per rendere l'applicazione sicura.

Minacce

Indisponibilità del servizio, mancato rispetto degli obblighi previsti dalla normativa vigente, perdita di efficacia o di efficienza nel tempo delle misure di sicurezza adottate.

A. ***Contromisure***Se ritenuto necessario, prima della fase di test e preferibilmente nella fase di sviluppo, effettuare opportune attività di code review in modo da eliminare potenziali problematiche di sicurezza, effettuando analisi:
   a. manuali sulla base di best practice (come ad esempio "OWASP - Code Review Guide");
   b. con strumenti automatici per l'analisi statica del codice (ad esempio strumenti free/open come ReviewPal per .NET e LAPSE+ per J2EE);
   c. con strumenti automatici per l'analisi dinamica del codice se disponibili.

   Le attività di code review devono essere comunque effettuate nel caso in cui il sistema gestisca dati giudiziari o nel caso in cui l'analisi del rischio abbia evidenziato uno livello qualitativamente alto del rischio non residuo.

B. Effettuare adeguati test e controlli di sicurezza sulle applicazioni prima della messa in produzione, anche in funzione del valore delle risorse da proteggere, utilizzando le liste di controllo "Test, deployment e gestione dell'applicazione" (Appendice B.2).

C. Non utilizzare dati di produzione in ambiente di test, in particolare nel caso in cui l'applicazione sia utilizzata per il trattamento di dati sensibili e/o giudiziari. Qualora sia necessario effettuare delle verifiche con dati "reali" utilizzare opportune tecniche di data masking, che offuschino i dati reali in maniera irreversibile per utilizzarli in ambienti di test.

D. In esercizio non esporre i servizi utente ed i servizi di gestione e di amministrazione sulle stesse interfacce, prevedere almeno la distinzione a livello 4 del modello ISO/OSI (per esempio differenti porte TCP) quanto non possibile a livello 3.

E. Documentare gli strumenti di gestione e di amministrazione dell'applicazione (per es. interfacce di configurazione, file di configurazione, ecc …).

F. In esercizio per le interfacce di gestione/amministrazione inibire l'utilizzo di credenziali di gruppo, ed utilizzare esclusivamente credenziali nominali.
   a. Effettuare, successivamente al deployment, adeguati controlli sulle applicazioni in produzione per assicurare l'efficienza e l'efficacia nel tempo dei meccanismi di sicurezza adottati.

G. Documentare, adeguare ed aggiornare nel tempo i meccanismi di sicurezza adottati, in funzione del valore dei dati e delle informazioni da proteggere e delle minacce di

sicurezza ad essi associati.

H. Utilizzare la lista di controllo "Test, deployment e gestione dell'applicazione" (Appendice B.2) come strumento di supporto per verificare la rispondenza dell'applicazione a quanto richiesto nei punti precedenti.

6. Requisiti minimi previsti dalla normativa vigente

Secondo quanto stabilito dalla regola 25 dell'Allegato B del D.Lgs. 196/03, qualora i destinatari del presente disciplinare siano soggetti esterni fornitori di prodotti o servizi utilizzati dall'Ente per l'adozione di misure minime di sicurezza ai sensi della normativa vigente, gli stessi devono attestare la conformità di quanto fornito alle seguenti disposizioni:

1) utilizzo di una procedura di autenticazione che permetta l'identificazione dell'incaricato attraverso opportune credenziali di autenticazione;
2) utilizzo di una parola chiave, quando prevista dal sistema di autenticazione, composta da almeno otto caratteri;
3) possibilità di modifica della parola chiave, quando prevista dal sistema di autenticazione, da parte dell'incaricato al primo utilizzo e, successivamente, almeno ogni sei mesi;
4) possibilità di disattivazione delle credenziali di autenticazione non utilizzate da almeno sei mesi, salvo quelle preventivamente autorizzate per soli scopi di gestione tecnica;
5) esistenza di meccanismi di autorizzazione per la separazione dei privilegi degli incaricati in base a diversi profili autorizzativi;
6) esistenza di meccanismi di backup che consentano il salvataggio dei dati con frequenza almeno settimanale.

Nel caso l'applicazione fornita sia destinata al trattamento di dati sensibili e/o giudiziari, l'attestazione deve inoltre indicare la conformità della stessa alle seguenti ulteriori disposizioni:

7) possibilità di modifica della parola chiave quando prevista dal sistema di autenticazione, da parte dell'incaricato al primo utilizzo e, successivamente, almeno ogni tre mesi;
8) esistenza di meccanismi di ripristino dei dati che permettano la ricostruzione degli stessi, in caso di danneggiamento, in tempi non superiori ai sette giorni;
9) utilizzo di tecniche di cifratura o codici identificativi, tali da rendere temporaneamente inintelligibili i dati sensibili e/o giudiziari anche a chi è autorizzato ad accedervi e da permettere l'identificazione degli interessati solo in caso di necessità.

I requisiti minimi sono vincolanti: non è possibile procedere con la messa in produzione di applicazioni per le quali non siano soddisfatti. L'owner dell'applicazione deve assicurarsi che venga rilasciata la conformità.

L’Appendice B.3 contiene una lista di controllo da utilizzare come strumento di supporto per verificare la rispondenza dell'applicazione ai requisiti minimi di sicurezza previsti dal D. Lgs. 196/03.

Appendice A - Esempio di analisi dei rischi per applicazioni web

1. Stabilire l'obiettivo di sicurezza

Rispondere ai requisiti minimi previsti dal Codice sulla protezione dei dati personali (D.Lgs. 196/03).

Garantire l'integrità dei dati, inseriti dagli utenti, successivamente alla pubblicazione sul web.

2. Descrivere sinteticamente l'applicazione

Applicazione web a tre livelli:

- *presentation:* pagine ASP.NET;
- *business logic:* class library C# + stored procedure;
- *data access:* SQL Server.

A livello infrastrutturale:

- l'applicazione è installata presso il CED della struttura
- la rete è protetta da firewall perimetrale che espone le sole porte 80 e 443
- i server (sviluppo, test e produzione) sono aggiornati periodicamente tramite sistemi di gestione automatica delle patch
- sui sistemi sono installati software antivirus aggiornati periodicamente
- le politiche di backup prevedono copia notturna dei dati di tipo incrementale

3. Minacce

Intercettazione delle credenziali di autenticazione

Intercettazione dei token o dei cookie di sessione

SQL Injection

Cross Site Scripting

4. Meccanismi di sicurezza da implementare

Autenticazione: permettere l'accesso ai dati alle sole persone autorizzate, tramite opportuna autenticazione (gli utenti si devono poter registrare sul web e la loro richiesta viene poi validata da un amministratore).

Autorizzazione: implementare due profili autorizzativi differenti: "operatore", "amministratore".

Gestione delle sessioni utente: prevedere meccanismi di cifratura dei token di sessione tramite funzioni di hash. Prevedere un time-out della sessione utente. Prevedere un meccanismo di forzatura del logout da parte dell'utente.

Validazione dei dati: effettuare controlli sui dati inseriti in input dall'utente e restituiti in output dall'applicazione. Utilizzare stored procedure per l'accesso ai dati di backend.

Gestione degli errori: l'applicazione è stata progettata prevedendo che, in caso di un'anomalia o di un errore software, il flusso operativo vada sempre a finire in uno stato "sicuro" (fail safe)

Tracciamento: non sono previsti meccanismi di logging di sicurezza, ad eccezione dei log di accesso riusciti/non riusciti.

Cifratura dei dati at rest ed in motion: non sono previsti meccanismi di cifratura dei dati in locale. E' prevista la cifratura del traffico nella fase di autenticazione tramite protocollo SSL.

Disponibilità dei dati: l'applicazione prevede dei meccanismi di backup dei dati con frequenza almeno settimanale

Documentazione di progetto: è stata redatta apposita documentazione che censisce, quanto più dettagliatamente possibile, i flussi informativi e le modalità di comunicazione con altri attori o sistemi

Codice applicativo: l'applicazione è stata sviluppata prestando attenzione a non inserire all'interno dei commenti del codice dati sensibili, critici, o che consentano di comprendere la logica dell'applicazione

Appendice B: Liste di controllo

Compilare le seguenti checklist riportando nelle colonne:

- Implementazione:
  - SI: Indica che il requisito è implementato;
  - NO: Indica che il requisito NON è implementato;
  - NA: Il requisito non è applicabile al contesto dell'applicazione;
- Note: Specificare eventuali note a supporto della dichiarazione.

Attenzione nel caso in cui alcune contromisure od attività (ad esempio: sistema di autenticazione, gestione dei log, SIEM, disponibilità dei dati, backup, ecc …) siano demandate ad altre applicazioni/sistemi centralizzati, riportare come stato implementato e nelle note indicare il sistema utilizzato ed eventuali configurazioni ad hoc richieste.

B.1 Design e sviluppo dell'applicazione

Analisi dei requisiti e design	Note di compilazione e risposte attese	Implementazione	Note
Nell'analisi dei requisiti è stato considerato il valore dei dati e delle informazioni trattate dall'applicazione	SI, indicare i macro dati identificati		
Tipologia di dati trattati dall'applicazione	Indicare un valore da 1 a 5 1 – L'applicazione non viene utilizzata per il trattamento di dati personali		

Analisi dei requisiti e design	Note di compilazione e risposte attese	Implementazione	Note
	2 – L'applicazione viene utilizzata pei li trattamento di dati personali solo di dipendenti interni 3 – L'applicazione viene utilizzata per il trattamento di dati personali relativi non solo a dipendenti interni (fornitori, utenti, ecc) 4 – L'applicazione viene utilizzata per il trattamento di dati personali sensibili 5 – L'applicazione viene utilizzata per il trattamento di dati personali, sensibili e giudiziari		
È stata eseguita l'analisi dei rischi incombenti sui dati	Esito dell'analisi e riferimenti documentali		
Sono stati considerati i vincoli architetturali e tecnologici imposti dall'infrastruttura esistente (servizi, porte, protocolli, tecnologie, ecc ...)	Elenco dei vincoli considerati		
Sono state documentate le porte ed i protocolli di comunicazione utilizzati dall'applicazione	SI/NO, censimento porte/protocollo		
Sono stati previsti meccanismi di autenticazione degli utenti	Obbligatorio per dati in perimetro d.lgs 196/2003, compilare sezione "Autenticazione"		
Sono stati previsti meccanismi di autorizzazione e profilatura utenti	Obbligatorio per dati in perimetro d.lgs 196/2003, compilare sezione "Autorizzazione"		

Analisi dei requisiti e design	Note di compilazione e risposte attese	Implementazione	Note
Sono stati previsti meccanismi di gestione sicura delle sessioni utente	SI nel caso sia prevista autenticazione, compilare sezione "Gestione delle sessioni utente"		
Sono stati previsti meccanismi di validazione dei dati in ingresso e in uscita	SI nel caso di gestione di contenuti dinamici da e verso altri sistemi/attori, compilare sezione "Validazione dei dati"		
Sono stati previsti meccanismi di gestione degli errori	SI nel caso di applicazioni enterprise, compilare sezione "Gestione degli errori"		
Sono stati previsti meccanismi di tracciamento	Obbligatorio per dati in perimetro d.lgs 196/2003, compilare sezione "Tracciamento"		
Sono stati previsti meccanismi di monitoraggio dell'applicazione	SI nel caso di applicazioni enterprise, compilare sezione "Monitoraggioi"		
Sono stati previsti meccanismi di cifratura dei dati	Obbligatorio per dati in perimetro d.lgs 196/2003, compilare sezione "Cifratura dei dati"		
Sono stati previsti meccanismi di disponibilità dei dati	Obbligatorio per dati in perimetro d.lgs 196/2003, compilare sezione "Disponibilita dei dati"		
È stata predisposta una documentazione di progetto	SI/NO/NA, in caso positivo compilare la sezione "Documentazione di progetto"		

Analisi dei requisiti e design	Note di compilazione e risposte attese	Implementazione	Note
Sono state previste metodologie per uno sviluppo sicuro del codice applicativo	SI/NO/NA, in caso positivo compilare la sezione "Codice applicativo"		
Sono state previste modalità per la sicurezza delle applicazioni su dispositivi mobili	SI/NO/NA, in caso positivo compilare la sezione "Applicazioni mobili"		

Autenticazione		Note di compilazione e risposte attese	Implementazione	Note
Sono stati definiti i punti di ingresso dell'applicazione che necessitano di meccanismi di autenticazione		Elenco delle macro aree/funzionalità applicative che richiedono autenticazione		
É stato scelto il meccanismo di autenticazione considerando anche i sistemi centralizzati già parte dell'infrastruttura esistente		Tipologia di autenticazione ed eventuali sistemi centralizzati utilizzati		
Sono stati rispettati i requisiti minimi obbligatori per legge nel caso di trattamento di dati personali:	disattivazione delle credenziali non utilizzate da almeno 180 giorni	Obbligatorio per dati in perimetro d.lgs 196/2003		
	non riutilizzo dei codici di identificazione già impiegati assegnandoli ad altri utenti (neanche in tempi diversi)	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare la tipologia di contromisura adotta		
In caso di autenticazione basata su userid+password, sono stati rispettati i requisiti minimi obbligatori per legge nel caso di trattamento di dati personali:	lunghezza minima consentita per la password di 8 caratteri	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare la lunghezza minima		
	scadenza della password non superiore ai 180 giorni (90 giorni nel caso di dati sensibili e/o	Obbligatorio per dati in perimetro d.lgs 196/2003		

Autenticazione		Note di compilazione e risposte attese	Implementazione	Note
	giudiziari)			
	possibilità per l'utente di modificare la propria password al primo login	Obbligatorio per dati in perimetro d.lgs 196/2003		
Sono stati previsti meccanismi di lockout di un account dopo *n* tentativi di accesso non riusciti		Obbligatorio per dati in perimetro d.lgs 196/2003, indicare il numero massimo di tentativi		
Sono stati previsti meccanismi di delay di n secondi a seguito di errata autenticazione		In caso positivo indicare il delay		
Sono stati previsti meccanismi di sblocco automatico del lockout di un account dopo n minuti		In caso positivo indicare il tempo di sblocco		
Sono stati previsti meccanismi di reset e modifica delle password da parte degli utenti senza l'intervento dell'amministratore di sistema		In caso positivo indicare la tipologia di meccanismo di sblocco		
Sono stati previsti meccanismi di cifratura delle password conservate in locale (hash)		Indicare se le password sono cifrate o conservate sotto forma di hash, in entrambi i casi riportare l'algoritmo utilizzato		
Sono stati previsti meccanismi di cifratura delle password trasmesse sulla rete		Indicare se le credenziali viaggiano su canale cifrato od in chiaro, ed eventuali meccanismi di cifratura (del canale o della password)		
Sono stati previsti meccanismi di controllo della robustezza delle password (regole di complessità)		Obbligatorio nel caso di trattamento di dati come definiti dal d.lgs 196/2003, riportare le specifiche per		

Autenticazione	Note di compilazione e risposte attese	Implementazione	Note
	le quali viene accettata una password (ad esempio: almeno 2 caratteri in maiuscolo, 2 numeri e 3 caratteri speciali)		
É stato ridotto al minimo il debugging remoto, in modo da fornire all'utente, in caso di errore di autenticazione, le sole informazioni indispensabili	Eventualmente riportare il messaggio unico mostrato all'utente		
In caso di autenticazione machine-to-machine e/o application-to-application sono state utilizzate tecniche di autenticazione forte mediante l'uso di certificati digitali e chiavi asimmetriche	In caso di certificato riportare la certificate authority (ad esempio: interna all'Ente o pubblica)		

Autorizzazione	Note di compilazione e risposte attese	Implementazione	Note
Sono stati previsti meccanismi di separazione dei privilegi in funzione del profilo utente	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare i profili considerati e le macro aree/funzionalità applicativi a cui possono accedere		
I permessi sono stati assegnati secondo il principio del "minimo privilegio"	SI		
L'accesso alle risorse di sistema è limitato ai soli account privilegiati	SI		
Il login applicativo non ha accesso diretto in scrittura alle tabelle dei database	SI		
È stata prevista la possibilità di configurare i profili autorizzativi	In caso di risposta positiva indicare anche le macro-configurazioni applicabili		
In caso di applicazioni web, esistono distinzioni fra aree ad accesso pubblico ed aree ad accesso riservato	In caso di risposta positiva indicare la suddivisione		

Autorizzazione	Note di compilazione e risposte attese	Implementazione	Note
	delle macro aree/funzionalità applicativi ad eccesso pubblico o ristretto		
Sono stati utilizzati i meccanismi nativi del framework su cui si basa l'applicazione per la gestione degli accessi utenti	In caso di risposta positiva indicare le funzionalità del framework utilizzate		

Gestione delle sessioni utente	Note di compilazione e risposte attese	Implementazione	Note
Sono stati previsti meccanismi di protezione delle credenziali utilizzate per il riconoscimento dell'utente dopo il login	SI/NO/NA, è comunque preferibili che le credenziali vengano sostituite da un identificativo di sessione dopo il login		
Esistono meccanismi di time-out delle sessioni	In caso di risposta positiva indicare il tempo di validità massima per una sessione, ed il tempo dopo il quale viene automaticamente chiusa in caso di inattività		
Il contenuto degli identificatori di sessione è cifrato	SI/NO/NA, è comunque preferibile l'utilizzo di token o parametri cifrati che solo lato server possono essere cifrati/decifrati		
Gli identificatori di sessione sono trasmessi su canali cifrati	SI, indicando il tipo di protocollo/algoritmo che realizza il canale cifrato		
Sono stati utilizzati, per la generazione degli identificativi di sessione, gli algo-	In caso di risposta positiva		

Gestione delle sessioni utente	Note di compilazione e risposte attese	Implementazione	Note
ritmi messi a disposizione dal framework od application server sul quale è stata sviluppata l'applicazione	indicare le funzionalità del framework utilizzate		
Esistono meccanismi di logout che permettono all'utente di forzare la chiusura di una sessione	SI/NO/NA		
Nel caso di sessioni web autenticate e di protocollo https, sono stati impostati a true gli attributi Secure e HttpOnly dei cookie	SI		

Validazione dei dati		Note di compilazione e risposte attese	Implementazione	Note
Sono stati definiti i punti di ingresso e di uscita dell'applicazione che richiedono controlli di validazione dei dati		SI, enucleare le macro categorie (ad esempio: form web, servizi RPC, REST, SOAP)		
Sono stati previsti più livelli di controllo di validazione dei dati		SI, enucleare i macro punti (ad esempio: lato web server, application server, logica applicativa, ecc ...)		
I dati sono validati in termini di:	formato	SI		
	sintassi	SI		
	dimensioni	SI		
Tutti i controlli di validazione sono effettuati lato server		SI		
I meccanismi di validazione sono stati scelti anche considerando gli strumenti già disponibili		SI/NO/NA		
I dati sono validati in base alla strategia di:	accettare esclusivamente i dati riconosciuti come validi	SI (prediligere rispetto alle altre soluzioni)		
	rigettare i dati riconosciuti come non validi	SI		
	modificare i dati non validi per ren-	SI (se implementata solo		

	derli validi	come misura aggiuntiva alle altre)		
Sono stati previsti ed implementati sia meccanismi di validazione dell'input che dell'output		SI/NA		
Sono stati considerati i rischi derivanti da attacchi di tipo SQL Injection		SI/NA		
Sono stati considerati i rischi derivanti da attacchi di tipo LDAP Injection		SI/NA		
Sono stati considerati i rischi derivanti da attacchi di tipo Cross Site Scripting		SI/NA		
Sono stati considerati i rischi derivanti da attacchi di tipo Cross Site Request Forgery		SI/NA		
Sono stati considerati altri rischi derivanti da problematiche note in letteratura (ad esempio vedi OWASP) e relative alla validazione dei dati		SI/NO, in caso di risposta affermativa indicare le problematiche considerate		
Sono stati previsti meccanismi di firma digitale e controllo della firma per i dati		SI nel caso in cui sia espressamente richiesto il non ripudio dei dati		

Gestione degli errori	**Note di compilazione e risposte attese**	**Implementazione**	**Note**
Sono stati previsti meccanismi che consentano di lasciare l'applicazione in uno stato sicuro (fail safe) in caso di errore	SI/NO, in caso di risposta positiva enucleare le strategie adottate		
Sono state configurate le funzionalità di logging/debugging per gli ambienti di esercizio in modo da tracciare solo le problematiche e non il dettaglio delle operazioni	SI/NO/NA, in caso positivo indicare le informazioni che si potrebbero ritrovare nei log		
Sono stati utilizzati linguaggi/framework di sviluppo che consentono la gestione strutturata degli errori	SI/NO/NA		

Tracciamento/Logging		Note di compilazione e risposte attese	Implementazione	Note
Sono stati definiti in fase di design gli eventi chiave per la sicurezza da rilevare tramite tracciamento/logging		SI/NO/NA		
Sono previsti meccanismi di rilevamento degli eventi applicativi di:	autenticazione applicativa (login e logout, riusciti e non)	Obbligatorio per dati in perimetro d.lgs 196/2003		
	accesso ai dati (lettura e scrittura)	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare i meccanismi associati alla tipologia di dati		
	modifica di funzioni amministrative	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare i meccanismi associati alle funzioni		
Esistono meccanismi di controllo e modifica del livello di granularità dei dati rilevabili		SI/NO/NA, in caso di risposta positiva indicare le granularità		
È prevista la possibilità di registrare, all'interno di un voce di log, le seguenti informazioni:	data ed ora dell'evento	Obbligatorio per dati in perimetro d.lgs 196/2003		
	luogo dell'evento (macchina/hostname, indirizzo IP)	Obbligatorio per dati in perimetro d.lgs 196/2003		
	identificativo dell'entità che ha generato l'evento (utente, servizio, processo)	Obbligatorio per dati in perimetro d.lgs 196/2003		
	descrizione dell'evento	Obbligatorio per dati in perimetro d.lgs 196/2003		
Sono state strutturate le informazioni nei log in modo tale che contengano solo informazioni per la ricostruzione di eventi o problematiche e non dati critici		SI/NO/NA, in caso di risposta negativa indicare eventuali dati critici che potrebbero essere presenti nei log		

Tracciamento/Logging		Note di compilazione e risposte attese	Implementazione	Note
Sono presenti meccanismi di interconnessione con il sistema regionale di gestione dei log (SIEM)		Obbligatorio per dati in perimetro d.lgs 196/2003		
I log sono prodotti nel formato Common Event Format (CEF)		SI/NO/NA, in caso di risposta negativa indicare il formato dei log		
In caso di impossibilità d'interconnessione con il SIEM, è prevista la conservazione dei log in file in cui sia possibile:	effettuare la scrittura incrementale o su supporti non riscrivibili	SI/NA		
	prevedere meccanismi di backup dei log secondo le procedure di backup centralizzato previste dall'Ente	SI/NA		
Esistono meccanismi di rotazione dei log		SI/NO/NA		
È possibile configurare la frequenza di rotazione dei log		SI/NO/NA, in caso di risposta positiva indicare la frequenza		
L'accesso ai log è consentito ai soli account privilegiati		Obbligatorio per dati in perimetro d.lgs 196/2003		
Esistono sistemi di verifica del funzionamento dei sistemi di logging		Obbligatorio per dati in perimetro d.lgs 196/2003		

Monitoraggio		Note di compilazione e risposte attese	Implementazione	Note
Sono stati definiti in fase di design gli eventi chiave per la sicurezza da monitorare		SI, enucleare le macro categorie di eventi		
Sono previsti sistemi di monitoraggio almeno per le seguenti informazioni:	stato dell'applicazione	SI/NA		
	utilizzo delle risorse ed altri dati che consentano un'efficiente/efficace fase di troubleshooting	SI/NA		

Monitoraggio		Note di compilazione e risposte attese	Implementazione	Note
	eventi di sicurezza individuati in fase di design e coerenti con quanto definito rispetto al tracciamento	SI/NA		
L'applicazione è stata progettata in maniera tale da prevedere meccanismi di interconnessione con la piattaforma di monitoraggio centralizzato dell'Ente.		SI/NA		

Crittografia dei dati	Note di compilazione e risposte attese	Implementazione	Note
È stata eseguita un'analisi dei rischi per valutare se utilizzare o meno meccanismi di cifratura dei dati	SI/NA, riportare l'esito dell'analisi e riferimenti documentali		
I dati sensibili/giudiziari/critici vengono mantenuti in chiaro solo per il tempo necessario alla loro elaborazione	SI/NA, in caso di risposta positiva indicare il tempo massimo in cui sono in chiaro		
I dati sensibili/giudiziari/critici sono conservati (data at rest) cifrati	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare le modalità di gestione		
I dati sensibili/giudiziari/critici sono trasmessi (data in motion) su canali cifrati	Obbligatorio per dati in perimetro d.lgs 196/2003, indicare le modalità di trasmissione		
I dati sensibili/giudiziari/critici sono trasmessi (data in motion) cifrati	SI/NO/NA		
Le password di autenticazione e chiavi private o di cifratura sono conservati (data at rest) cifrati	SI, indicare le modalità di gestione		
Le password di autenticazione e chiavi private o di cifratura sono trasmessi	SI, indicare le modalità di		

Crittografia dei dati		Note di compilazione e risposte attese	Implementazione	Note
(data in motion) su canali cifrati		trasmissione		
Le password di autenticazione e chiavi private o di cifratura trasmessi (data in motion) cifrati		SI/NO/NA		
Sono stati previsti meccanismi di firma digitale nei casi in cui sia necessario garantire la non ripudiabilità		SI/NA, in caso di risposta affermativa indicare i meccanismi adottati e le eventuali certificate authority utilizzate		
Le chiavi utilizzate per la firma digitale sono distinte dalle chiavi di cifratura		SI/NA		
Sono utilizzati certificati rilasciati/riconosciuti dall'Ente (trusted Certificate Authority) per i flussi dati all'interno dell'Ente stesso		SI/NO/NA, in caso di risposta positiva indicare la Certificate Authority		
Sono utilizzati certificati rilasciati da Certificate Authority attendibili e riconosciute a livello nazionale/internazionale per i flussi dati diretti o provenienti dall'esterno dell'Ente		SI/NO/NA, in caso di risposta positiva indicare la Certificate Authority		
Viene verificata tutta la catena dei certificati almeno in caso di mutua autenticazione, o di connessione verso servizi esterni		SI/NA		
Sono stati scelti algoritmi crittografici (cifratura simmetrica, asimmetrica e funzioni di hash) standard e robusti, con chiavi aventi lunghezza adeguata		SI/NA, in caso di risposta positiva indicare gli algoritmo selezionati, in caso di risposta negativa riportare le motivazioni (ad esempio compatibilità sistemi legacy)		
Sono stati previsti adeguati meccanismi di gestione sicura delle chiavi:	distribuzione	SI/NO/NA		
	conservazione	SI/NA		
	riciclo periodico	SI/NO/NA		
	revoca	SI/NO/NA		

Crittografia dei dati		Note di compilazione e risposte attese	Implementazione	Note
	recovery	SI/NA		
	distruzione	SI/NO/NA		
Nel caso in cui l'applicazione risieda su sistemi virtualizzati è stata valutata la possibilità di cifrare sempre e comunque lo spazio disco su cui risiedono i dati		SI/NO/NA		

Disponibilità dei dati	Note di compilazione e risposte attese	Implementazione	Note
È stato definito, in fase di design, l'indice di criticità operativa. Se sì indicarne il valore.	**Inserire uno dei seguenti valori:** 1- Non causa particolari danni sull'operatività della struttura erogante 2- Potrebbe ostacolare l'operatività senza ripercussioni sui servizi erogati e senza sanzioni amministrative 3 - Blocca alcuni processi aziendali con possibili ripercussioni dirette sui servizi erogati e senza sanzioni amministrative 4 - Blocca alcuni processi aziendali con ripercussioni dirette sui servizi erogati o con possibili sanzioni amministrative 5 - Blocca alcuni processi aziendali con importanti ri-		

Disponibilità dei dati	Note di compilazione e risposte attese	Implementazione	Note
	percussioni dirette sui servizi erogati o con sanzioni amministrative		
I meccanismi di backup utilizzano punti di sincronizzazione aperti e compatibili con le policy di backup dell'Ente	SI/NO/NA		
Sono necessarie eccezioni rispetto alle policy di backup dell'Ente	**SI/NO/NA** Le policy di backup dell'Ente sono dipendenti dalla filiera utilizzata. Per tutte le filiere la politica di backup dei file su filesystem è di tipo incrementale. Vengono tenute 15 versioni del file se il file esiste. Ogni versione è mantenuta 60 giorni poi viene cancellata. La durata dell'ultimo backup eseguito è a vita se il file non viene cancellato, altrimenti è di 180 gg. Tale backup è schedulato tutte le notti. Le policy di backup per i dati sono le seguenti. **Filiera A**: Il backup del database ORACLE di Produzione avviene con il TDP per Oracle di Tivoli. La politica di backup prevede le seguenti schedulazioni auto-		

Disponibilità dei dati	Note di compilazione e risposte attese	Implementazione	Note
	matiche: - 2 o 3 backup completi on-line settimanali con reten-tion 30 giorni - 6 backup giornalieri degli archive log Per tutti i DB Oracle è schedulato anche un ex-port notturno giornaliero, Ogni versione è mantenuta per 20 giorni poi viene can-cellata. La durata dell'ulti-mo backup eseguito è a vita se il file non viene can-cellato, altrimenti è di 20 giorni. Mensilmente viene fatto un Export che viene tenuto per 12 mesi. **Filiera B**: Il backup del database MSSQL di produzione av-viene con il TDP per MS-SQL di Tivoli. La politica di backup prevede le seguen-ti schedulazioni automati-che: - 1 backup completo online giornaliero con retention 30 giorni - 6 backup giornalieri degli		

Disponibilità dei dati	Note di compilazione e risposte attese	Implementazione	Note
	archive log Mensilmente viene fatto un Backup che viene tenuto per 12 mesi. **Filiera C**: Per tutti i DB MySQL e PostgreSQL è schedulato un processo che genera su file system il dump del database. Alla notte è attiva una schedulazione Tivoli che salva su TSM Server i backup prodotti su file system. Ogni versione è mantenuta per 20 giorni poi viene cancellata. La durata dell'ultimo backup eseguito è a vita se il file non viene cancellato, altrimenti è di 20 giorni. **Se SI, indicare qual è l'eccezione necessaria**		
Esistono meccanismi di backup che consentono il salvataggio dei dati con frequenza almeno settimanale	Rispondere solo se non viene utilizzato il sistema di backup centralizzato Obbligatorio per dati in perimetro d.lgs 196/2003 (dati personali)		
Esistono meccanismi che consentono il ripristino dei dati in tempi inferiori ai	Rispondere solo se non		

Disponibilità dei dati	Note di compilazione e risposte attese	Implementazione	Note
sette giorni	viene utilizzato il sistema di backup centralizzato Obbligatorio per dati in perimetro d.lgs 196/2003 (dati personali)		

Documentazione di progetto	Note di compilazione e risposte attese	Implementazione	Note
La documentazione è stata redatta utilizzando standard internazionali	SI, in caso di risposta negativa indicare gli standard utilizzati		
È presente apposita documentazione che censisca i flussi informativi e le modalità di comunicazione con altri attori o sistemi	SI, indicare riferimenti documentali		
È presente apposita documentazione sulle procedure di Continuità Operativa	SI, indicare riferimenti documentali		
È presente apposita documentazione su come procedere nel caso di Disaster Recovery	SI, indicare riferimenti documentali		
È presente apposita documentazione per la gestione degli errori applicativi e dell'applicativo	SI, indicare riferimenti documentali		

Codice applicativo	Note di compilazione e risposte attese	Implementazione	Note
In caso di codice sorgente che può essere visualizzato dall'utente finale, sono state adottate tecniche per prevenire che nel codice applicativo siano presenti dati sensibili/critici	SI/NA		
Sono state adottate tecniche per prevenire che all'interno del codice siano presenti credenziali/chiavi/passphrase, soprattutto in caso di codice sorgente che può essere visualizzato dall'utente finale	SI/NA		
Sono state previste delle verifiche (code review) sul codice per la prevenzione	SI, indicare le attività svol-		

Codice applicativo	Note di compilazione e risposte attese	Implementazione	Note
di bug	te		
Il codice delle applicazioni è stato opportunamente offuscato al rilascio in ambienti di esercizio	SI/NA, in caso di risposta affermativa indicare gli strumenti utilizzati		

Applicazioni per dispositivi mobili		Implementazione	Note
Le credenziali/token e le chiavi di cifratura sono memorizzate in aree del dispositivo mobile accessibili soltanto all'applicazione	SI		
I dati sensibili/giudiziari/critici non devono essere salvati nei supporti rimovibili del dispositivo mobile	SI/NA		
L'applicazione mobile è stata progettata per individuare se il dispositivo mobile su cui è stata installata è manomesso (Jailbreak/root)	SI/NA		
Sono state adottate tecniche per prevenire che nel codice applicativo dell'applicazione mobile siano presenti certificati per la validazione dell'identità del server	SI/NA		
L'applicazione mobile verifica tutta la catena dei certificati (certificate trust chain) fornita dal server prima di instaurare la connessione	SI/NA		
Sono stati previsti meccanismi di validazione dei dati provenienti dall'applicazione mobile (controllo lato server)	SI/NA		
Le richieste di accesso si limitano a quelle indispensabili per il funzionamento dell'applicazione	SI/NA		
Elenco delle richieste di accesso necessari all'applicazione	Se la risposta alla domanda precedente è SI, elencare le richieste di accesso necessarie all'applicazione		
L'applicazione mobile è stata progettata per non utilizzare il caricamento dinamico del codice dall'esterno dell'applicazione	SI/NA		

Applicazioni per dispositivi mobili		Implementazione	Note
Sono stati previsti ed implementati meccanismi di verifica dei dati, sia in input che in output, sull'applicazione mobile (controllo lato applicazione)	SI/NA		

B.2 Test, deployment e gestione dell'applicazione

Test		Implementazione	Note
Sono stati eseguiti i controlli sul codice (code review)	Necessario in caso di che l'applicazione tratti dati giudiziari, od l'analisi del rischio abbia individuato livelli qualitativamente alti del rischio non residuo. SI/NA in caso di risposta positiva indicare se manuali, automatici statici o automatici dinamici		

Deployment		Implementazione	Note
Sono stati esposti su differenti interfacce di rete i servizi utente, di gestione, di amministrazione	SI/NA, indicare a quale livello del modello ISO/OSI sono stati differenziati		
In caso di ambienti virtualizzati, sono state utilizzate differenti interfacce di rete, e subnet, per i servizi utente, di gestione, di amministrazione	SI/NA		
Sono stati documentati i meccanismi di sicurezza adottati	SI, riportare riferimento documentale		
Sono state rispettate le policy centralizzate relativamente a porte, protocolli e servizi utilizzabili	SI/NA		
È stata fornita dal fornitore esterno l'attestazione di conformità alle misure minime di sicurezza previste dalla legge	SI/NA, in caso di risposta affermativa riportare riferimento documentale		

Gestione		Implementazione	Note
In ambiente di test non sono utilizzati dati di produzione	SI/NA		
In caso di verifiche con dati di produzione sono state adottate opportune tecniche di data masking	SI/NA, in caso di risposta positiva indicare le tecniche di data masking indicate, in caso di risposta negativa indicare la modalità di gestione dell'attività		
In caso di ambienti virtualizzati, sono stati previsti meccanismi per la continuità operativa, disaster recovery, backup dell'applicazione che non si basino esclusivamente sull'infrastruttura di virtualizzazione	SI/NO/NA		
Sono stati documentati gli strumenti di gestione e di configurazione dell'applicazione	SI/NA, in caso di risposta affermativa riportare riferimento documentale		
Sono stati previsti adeguati meccanismi di controllo degli accessi agli strumenti di gestione e configurazione	SI/NA, in caso di risposta affermativa riportare i controlli o riferimento documentale		
La connessione agli strumenti di gestione e configurazione avviene su canali cifrati	SI/NO/NA, in caso di risposta negativa indicare i servizi non cifrati		
Sono utilizzate esclusivamente credenziali nominale per l'accesso alle interfacce di gestione ed amministrazione	SI, in caso di risposta negativa indicare contromisure adottate		

B.3 Requisiti minimi previsti dalla normativa vigente

Misure minime da osservare per tutti i trattamenti		Implementazione	Note
Esiste una procedura di autenticazione che permette l'identificazione univoca dell'utente attraverso opportune credenziali di autenticazione	Obbligatorio per dati in perimetro d.lgs 196/2003		
È utilizzata una parola chiave (password), quando prevista dal sistema di autenticazione, composta da almeno otto caratteri	Obbligatorio per dati in perimetro d.lgs 196/2004		
Esiste la possibilità di modifica della parola chiave, quando prevista dal sistema di autenticazione, da parte dell'utente al primo utilizzo e, successivamente, almeno ogni sei mesi	Obbligatorio per dati in perimetro d.lgs 196/2005		
Esistono meccanismi di disattivazione delle credenziali di autenticazione non utilizzate da almeno sei mesi, salvo quelle preventivamente autorizzate per soli scopi di gestione tecnica	Obbligatorio per dati in perimetro d.lgs 196/2006		
I codici di identificazione già impiegati non sono riutilizzati nel tempo assegnandoli ad altri utenti	Obbligatorio per dati in perimetro d.lgs 196/2007		
Esistono meccanismi di autorizzazione per la separazione dei privilegi degli incaricati in base a diversi profili autorizzativi	Obbligatorio per dati in perimetro d.lgs 196/2008		
Esistono meccanismi di protezione dei dati contro le minacce di intrusione e dell'azione di programmi malevoli (es. cifratura delle password, impiego di firewall o di software antivirus, hardening dei sistemi, ecc.)	Obbligatorio per dati in perimetro d.lgs 196/2009		
I programmi sono aggiornati periodicamente per prevenire le vulnerabilità e correggerne difetti (es. patch di sistema, aggiornamenti antivirus, ecc.)	Obbligatorio per dati in perimetro d.lgs 196/2010		
Esistono meccanismi di backup e ripristino, con salvataggio dei dati effettuato con frequenza almeno settimanale	Obbligatorio per dati in perimetro d.lgs 196/2011		

Misure minime ulteriori da osservare nel caso di trattamenti di dati sensibili e/o giudiziari		Implementazione	Note
Obbligatorio per dati in perimetro d.lgs 196/2011	Obbligatorio per dati in perimetro d.lgs 196/2011		
Obbligatorio per dati in perimetro d.lgs 196/2011	Obbligatorio per dati in perimetro d.lgs 196/2011		
Obbligatorio per dati in perimetro d.lgs 196/2011	Obbligatorio per dati in perimetro d.lgs 196/2011		

Appendice C: Glossario

Applicazione web: applicazione client/server che interagisce con l'utente (o con altri sistemi) tramite protocollo HTTP.

Bruteforce (attacco a forza bruta): tecnica di attacco per l'individuazione di una password attraverso tentativi successivi di tutte le possibili combinazioni di caratteri. Il limite degli attacchi di tipo bruteforce risiede nel tempo necessario per portarli a termine.

Buffer overflow: tecnica di attacco basata sull'alterazione del normale flusso di esecuzione di un'applicazione mediante la sovrascrittura di aree di memoria riservate.

Code injection: tecnica di attacco che consiste nell'aggiunta di codice imprevisto all'interno di un'applicazione o di un sito web, al fine di alterarne il funzionamento.

Code tampering: tecnica di attacco che consiste nella modifica di codice non autorizzata di un'appicazione, al fine di alterarne il funzionamento.

Cross-site-scripting (XSS): tecnica di attacco che sfrutta un sito web vulnerabile per indirizzare codice maligno (iniettato dall'attaccante) verso il browser dell'utente vittima. Tale codice è eseguito sulla macchina dell'utente vittima dell'attacco.

Cross-site request forgery (CSRF/XSRF): tecnica di attacco che sfrutta un sito web dinamico vulnerabile consentendo al server di accettare richieste dell'utente senza che queste siano effettivamente state fatte in maniera intenzionalmente dall'utente.
Differentemente dall'XSS la vittima è il server che ripone troppa fiducia nelle richieste provenienti dei client/browser.

Data masking: è una tecnica di offuscamento dei dati che prevede un processo irreversibile per il quale i dati vengono resi non riconoscibili (de-identify) ma mantengono la consistenza originale.

Dati giudiziari: i dati personali idonei a rivelare provvedimenti di cui all'articolo 3, comma 1, lettere da a) a o) e da r) a u), del D.P.R. 14 novembre 2002, n. 313, in materia di casellario giudiziale, di anagrafe delle sanzioni amministrative dipendenti da reato e dei relativi carichi pendenti, o la qualità di imputato o di indagato ai sensi degli articoli 60 e 61 del codice di procedura penale.

Dati personali: qualunque informazione relativa a persona fisica, identificati o identificabili, anche indirettamente, mediante riferimento a qualsiasi altra informazione, ivi compreso un numero di identificazione personale.

Dati sensibili: i dati personali idonei a rivelare l'origine razziale ed etnica, le convinzioni religiose, filosofiche o di altro genere, le opinioni politiche, l'adesione a partiti, sindacati, associazioni od organizzazioni a carattere religioso, filosofico, politico o sindacale, nonché i dati personali idonei a rivelare lo stato di salute e la vita sessuale.

Denial-of-service (negazione del servizio): tecnica di attacco che limita o interrompe la disponibilità di un servizio rendendolo inaccessibile ai legittimi utilizzatori in modo temporaneo o permanente.

Funzioni di hash: funzioni matematiche che comprimono i bit di un messaggio digitale in un'impronta di dimensioni fisse (c.d. *hash*), in modo che a messaggi diversi corrispondano impronte diverse. Tali funzioni sono irreversibili, per cui dall'impronta non è possibile

ricavare il messaggio digitale che l'ha originato. *Hash:* impronta di un messaggio digitale calcolata tramite una funzione di hashing.

Incaricato: la persona fisica autorizzata a compiere operazioni di trattamento di dati personali.

Interessato: la persona fisica cui si riferiscono i dati personali.

Man in the middle (MitM) attack: situazione nella quale un attaccante è in grado di leggere, inserire o modificare a piacere i dati di una comunicazione fra due o più attori senza che questi ne siano a conoscenza.

Meccanismo di sicurezza: elemento di un sistema progettato per preservare la confidenzialità, l'integrità e la disponibilità delle risorse del sistema stesso.

Normativa vigente: il "Codice in materia di protezione dei dati personali", Decreto Legislativo numero 196 del 30 Giugno 2003, entrato in vigore il 1° Gennaio del 2004 (c.d. "Codice della Privacy").

OWASP: l'Open Web Application Security Project è un progetto open-source per la sicurezza delle applicazioni, fornisce sia indicazioni per le verifiche di sicurezza che per lo sviluppo sicuro;

Privilege escalation (scalata dei privilegi): tecnica di attacco utilizzata per eseguire azioni con permessi superiori a quelli posseduti. Esempio nei sistemi Windows based: accesso di un utente appartenente al gruppo *Users* a risorse riservate al gruppo *Administrators*.

Session hijacking (dirottamento di sessione): tecnica di attacco basata sull'intercettazione di una sessione al fine di accedere a risorse senza disporre dei permessi necessari. Esempio in un'applicazione web: intercettazione di una sessione utente per accedere ad un'area web ad accesso riservato.

SIEM (Security Information and Event Management): sistema di gestione delle informazioni di sicurezza e degli eventi di sicurezza.

Spoofing (Impersonificazione): tecnica di attacco che consente ad un'identità di impersonare, in modo illegittimo, un'entità differente.

SQL Injection: tecnica di attacco che consente l'esecuzione, sul database vittima, di query SQL costruite ad-hoc a partire dall'input utente non opportunamente filtrato.

Superficie di attacco: insieme dei potenziali punti di accesso ad un sistema che possono essere sfruttati da un attaccante per comprometterne la sicurezza.

REGIONE EMILIA-ROMAGNA
Atti amministrativi

GIUNTA REGIONALE

Grazia Cesari, Responsabile del SERVIZIO SISTEMA INFORMATIVO - INFORMATICO REGIONALE, in qualità di Responsabile della Sicurezza della Giunta regionale, esprime, ai sensi della deliberazione della Giunta Regionale n. 2416/2008 e s.m.i., parere di regolarità amministrativa in merito all'atto con numero di proposta DPG/2014/4203

data 20/03/2014

IN FEDE

Grazia Cesari