consip

Linee Guida di Programmazione
-Sicurezza delle applicazioni Web-

Versione 2.0

16 marzo 2010

DATA	VERSIONE	DESCRIZIONE	CAP. /SEZ. MODIFICATI
Febbraio 2004	1.0	Nascita del documento	Tutti
Marzo 2010	2.0	Integrazione con contromisure Owasp,	Modificati tutti i capitoli e aggiunto il Cap. 5

Storia del documento

	DATA	NOMINATIVO	AREA/FUNZIONE	DIREZIONE
Redatto da:	01/02/2010	Raffaella Migliorini	System Solution	DINIT
		Luana Angelone	System Solution	DINIT
Verificato da:	08/02/2010	Angelo Stati	System Solution	DINIT
Approvato da:	26/02/2010	Marina Venzo	Standard e Sistemi Informativi Interni	DBS

Condivisione del documento

Sommario

Indice delle figure

1. INTRODUZIONE

1.1 Scopo e campo di applicazione

Obiettivo del presente documento è dare un insieme di requisiti ed indicare contromisure, in linea con gli standard internazionali, che hanno lo scopo di mitigare il rischio di un sistema Informativo esposto sulla intranet o su Internet. Queste linee guida di programmazione sono comprensive degli aspetti infrastrutturali a supporto e completamento alla sicurezza delle applicazioni e sono corredate dalle possibili contromisure applicabili.

1.2 Contesto di riferimento e Modalità operativa

Le attuali tendenze dello sviluppo delle applicazioni web del MEF richiedono una sempre maggior attenzione agli aspetti di sicurezza nello sviluppo ed utilizzo delle stesse. Infatti, negli ultimi anni, ad uno scenario d'applicazioni esclusivamente ad uso interno (dominio Tesoro) si è progressivamente sostituito uno scenario d'applicazioni che offrono servizi all'esterno sia ad utenza infranet (SPC) sia ad utenza internet. Questo, da un certo punto di vista, significa che le applicazioni vanno a posizionarsi sul perimetro esterno della rete MEF e quindi, se non correttamente sviluppate potrebbero costituire una breccia.

Lo sviluppo sicuro del codice di una applicazione Web e la relativa review deve seguire delle fasi ben precise appoggiandosi a delle metodologie consolidate. Tra le metodologie prese in esame, per le applicazioni MEF, è stata adottata quella che attualmente si presenta come uno standard de-facto sul mercato internazionale, sia americano che europeo: l'OWASP, Open Web Application Security Project che mira a standardizzare tutte le fasi di progettazione, manutenzione e revisione di un'applicazione.

Le indicazioni descritte nel presente documento devono considerarsi di riferimento sia per lo sviluppo di applicazioni WEB "critiche" che "non critiche" secondo la classificazione descritta nel documento "Linee guida per la definizione della criticità di applicazioni Web e per lo sviluppo di applicazioni critiche".

Si vuole però sottolineare che la sicurezza delle applicazioni web deriva da una corretta valutazione degli elementi di sicurezza in ogni fase del processo di sviluppo. Infatti sia che il processo di sviluppo utilizzi il modello "extreme programming" (XP), quello iterativo o quello "waterfall", di fatto, questi modelli sono tutti, in maniera più o meno marcata, riconducibili alle fasi rappresentate nella figura seguente.

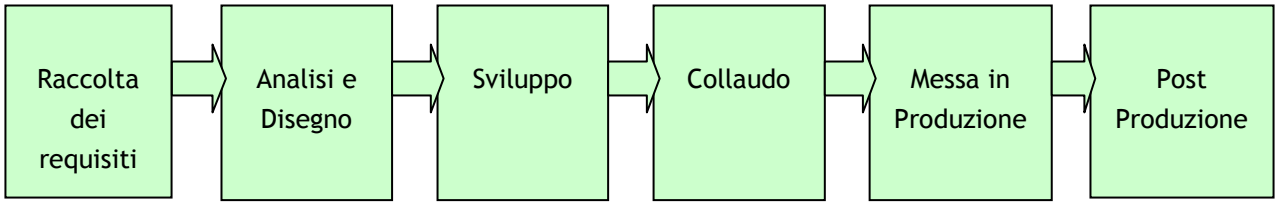

Figura 1 – Fasi dello sviluppo di una applicazione

La messa in sicurezza di un'applicazione web ha inizio già durante la fase di raccolta dei requisiti con l'individuazione dei requisiti utente o "Use Case di business", descrittivi dei processi funzionali di un sistema. Ogni "Use Case" deve contenere le informazioni necessarie per rispondere alla domanda: "quale processo assicura che i dati del sistema e le informazioni utente sono sicure?" e questo si traduce nel comprendere la sicurezza nella definizione dei requisiti funzionali o non funzionali di un'applicazione.

Accanto alla necessità di prevedere la modellazione dei processi funzionali legati alla sicurezza, devono essere descritte, per poi essere sviluppate, le regole che descrivono aspetti funzionali che non possono essere rappresentati come processi ad esempio la disattivazione di un'utenza dopo un periodo d'inattività.

La sicurezza delle applicazioni web è completata poi dai vincoli tecnici; tali vincoli non derivano da necessità utente ma sono definiti dalle caratteristiche delle infrastrutture sottese alle applicazioni e dalle politiche di sicurezza impostate. Un esempio di vincolo tecnico può essere "Apertura delle sole porte 80 e 443 sulla web farm" oppure sempre a livello di web farm il timeout forzato delle sessioni dopo 10 minuti.

In modo analogo, nella fase di analisi e disegno, quando vengono sviluppati i diagrammi delle classi e delle interazioni, è necessario che vengano inseriti "Security Object" che contengano attributi e metodi che implementino le contromisure indicate ad esempio il blocco di dati di input qualora questi non rispettino il formato previsto.

L'utilizzo di linee guida nella sola fase di sviluppo può non essere sufficiente se non verificata in fase di test/collaudo. Per tutte le applicazioni dovrà essere redatto il piano di test secondo le "Linee guida per il piano di test relativo alla sicurezza delle applicazioni web".

Il documento è articolato in cinque capitoli.
Nel primo è descritto lo scopo e l'obiettivo del documento; il secondo capitolo fornisce lo schema architetturale di riferimento; il terzo descrive l'ambiente "run-time" in cui le applicazioni MEF sono eseguite; il quarto descrive quegli elementi che, indipendentemente dal linguaggio di programmazione (Java .NET etc.), concorrono a costruire la sicurezza di un'applicazione ed il quinto indica le contromisure che devono essere messe in atto.

2. SCHEMA ARCHITETTURALE DI RIFERIMENTO

La figura che segue riporta quella che è oggi nel MEF l'infrastruttura di riferimento a supporto della maggior parte delle applicazioni a tre livelli e quindi dei siti ad esse sottesi. Per semplicità di lettura l'infrastruttura rappresentata è quella esposta verso Internet ma i siti interni sono dotati di una topologia del tutto simile.

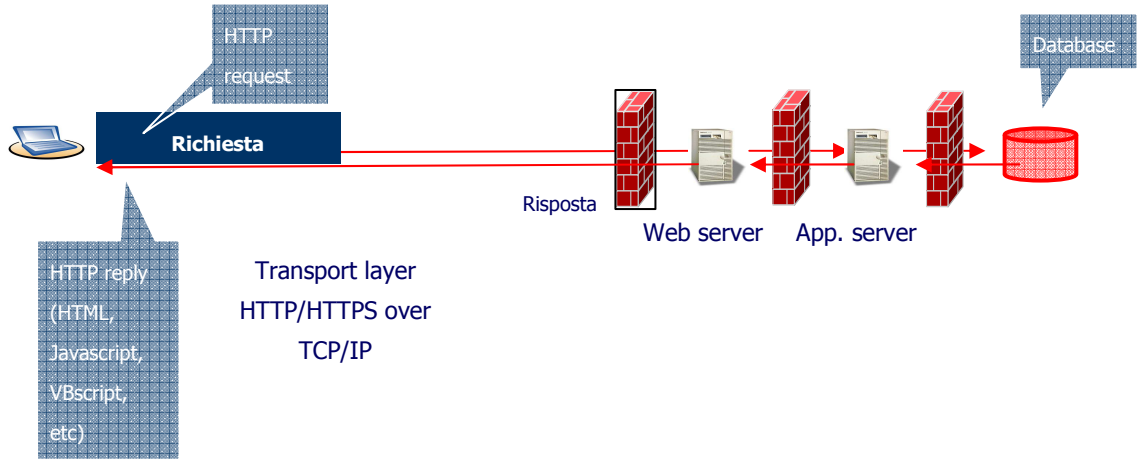

Figura 2 – Architettura di riferimento

I web server sono organizzati in "web farm" gestite da un dispositivo di bilanciamento del traffico HTTP. Ciascuna farm è costituita da più macchine fisiche ma si presenta con un unico indirizzo IP detto Virtual IP (VIP).
L'impianto prevede che non sia presente codice applicativo a questo livello, fa eccezione per caratteristiche architetturali la farm .NET che prevede la presenza di parte del framework ASP.NET nella componente listener.

Gli Application Server, così come i Database Server (o altre fonti di dati), sono protetti da un ulteriore livello di firewall e non sono accessibili direttamente dall'utenza.

3. AMBIENTE RUN-TIME

In questo capitolo si vogliono accennare alcune caratteristiche che devono essere presenti nell'ambiente run-time di un'applicazione sia esso quello di collaudo o di esercizio.

È infatti necessario assicurare a livello di infrastruttura che l'ambiente in cui l'applicazione è testata (ambiente di collaudo) abbia tutte le caratteristiche di sicurezza dell'ambiente di esercizio questo al fine di evitare la segnalazione di vulnerabilità legate all'ambiente e non a bachi dell'applicazione.

È necessario quindi che i server siano installati e configurati in maniera sicura con procedure documentate e periodicamente controllate.
Tra le procedure indispensabili si individuano:

- piano di backup e recovery;
- limitazione degli accessi al server ed il cambio periodico delle password di amministratore;
- chiusura delle porte dei Web server ad esclusione di quelle necessarie alle applicazioni (80 e 443);
- corretta definizione e segmentazione delle connessioni di rete;
- applicazione costante delle patch ai sistemi operativi;
- eliminazione di tutti i file obsoleti dai web server;
- eliminazione di tutti i servizi non necessari dai web server.

Si sottolinea che non vuole essere una trattazione esaustiva ma piuttosto un'indicazione di quali accortezze sono messe in opera per garantire la sicurezza di un servizio/applicazione. Per maggiori dettagli si rimanda ai documenti di configurazione/installazione esistenti.

3.1 Configurazione sistema operativo

Un primo elemento di attenzione è riservato all'installazione dei sistemi operativi (Windows/Unix).
In fase di installazione dei server vengono osservate le seguenti regole:

- il sistema operativo viene installato al livello consolidato di patch di sicurezza mantenuto presso il CED di appartenenza, ed inserito nel processo di aggiornamento periodico delle stesse
- gli account applicativi o di sistema, non usati, sono rinominati così come gli account di default
- sono attivati i meccanismi di auditing in modo da registrare il verificarsi di eventi significativi dal punto di vista della sicurezza. Gli eventi registrati includono:
    - log-on e log-off
    - tentativi di accesso al sistema riusciti e falliti

    - tentativi di accesso a risorse e dati riusciti e falliti
    - avvio e arresto del sistema
    - avvio e arresto delle funzioni di audit
    - la connessione e disconnessione dei device di input/output
- la registrazione deve riportare almeno i seguenti dati:
    - identità dell'utente che ha scatenato l'evento
    - data e ora dell'evento
    - tipo dell'evento
    - oggetti coinvolti dall'evento (file, applicazioni, ecc.).
- i dati relativi agli eventi registrati sono conservati per un periodo di tempo sufficiente alla loro analisi e/o utilizzazione a fini statistici, e comunque non inferiore ai 6 mesi, come prove da esibire in caso di dispute, come elementi da considerare nell'identificazione di misure migliorative della sicurezza
- i servizi ed i protocolli non necessari sono disattivati
- su alcuni filesystem sono applicate ACL più restrittive rispetto a quelle di default
- viene installato l'antivirus (se piattaforma Windows) ed il server è inserito nell'infrastruttura Enterprise di aggiornamento.

Una volta installato ogni server diventa oggetto di aggiornamenti periodici delle patch di sicurezza e di controlli di Vulnerability Check.

3.2 Web Server

Come descritto in precedenza i web server sono organizzati in web farm. La configurazione dei listener (Apache o IIS) prevede, come passi minimali:

- l'attivazione dei log di errore e di accessi nel formato WSC (per questi ultimi ai fini statistici è in corso di realizzazione un'infrastruttura di raccolta e trattamento basata sul prodotto WebTrends)
- la rimozione di tutte le applicazioni/servizi demo, nonché ogni altro servizio, utility di sistema o funzionalità non strettamente necessaria
- la disattivazione degli account di default
- la configurazione del controllo d'accesso per gli utenti anonimi in base ai seguenti criteri:
    - abilitare l'accesso sui contenuti statici (.txt, .gif, .jpg, .html) in sola lettura
    - abilitare l'accesso agli eseguibili, fruibili via web, in sola esecuzione
    - abilitare l'accesso agli script (.asp, .php, ecc.), fruibili via web, in sola esecuzione
    - abilitare l'accesso ai file di tipo 'include' (.inc, .shtm, ecc.), fruibili via web, in sola esecuzione
- l'attivazione, se richiesto, del supporto https con l'installazione dell'apposito certificato.

Nel caso di siti istituzionali a contenuto informativo è poi opportuno:

- che siano definite procedure formali per la pubblicazione di informazioni su siti web che prevedano:
  - controllo dell'attendibilità delle fonti da cui sono state tratte le informazioni
  - protezione delle informazioni nel caso di memorizzazione su supporti temporanei precedenti alla pubblicazione
  - workflow formalizzato per la pubblicazione delle pagine web
  - verifica periodica dell'attendibilità delle pagine pubblicate
  - eventualmente l'utilizzo di strumenti per la salvaguardia dell'integrità delle informazioni pubblicate (ad esempio, la firma delle pagine web).

3.3 Application Server

In modo analogo a quanto previsto per i Web Server anche l'installazione delle piattaforme Application Server richiede le seguenti accortezze:

- l'attivazione dei log di piattaforma/sottosistema per il tracciamento degli errori (non di applicazione)
- la rimozione di tutte le applicazioni/servizi demo, nonché ogni altro servizio, utility o funzionalità non strettamente necessaria
- la disattivazione degli account di default.

Per queste piattaforme risulta particolarmente importante la restrizione delle autorizzazioni all'interfaccia amministrativa, che deve essere acceduta solo con autenticazione e da un ristretto numero di utenti appartenenti esclusivamente alla struttura di gestione sistemistica e possibilmente solo dalla rete locale.

4. LINEE GUIDA DI PROGRAMMAZIONE

La figura, che segue, descrive la sicurezza di un'applicazione web come il risultato delle misure di sicurezza applicate su ogni livello fisico e logico.

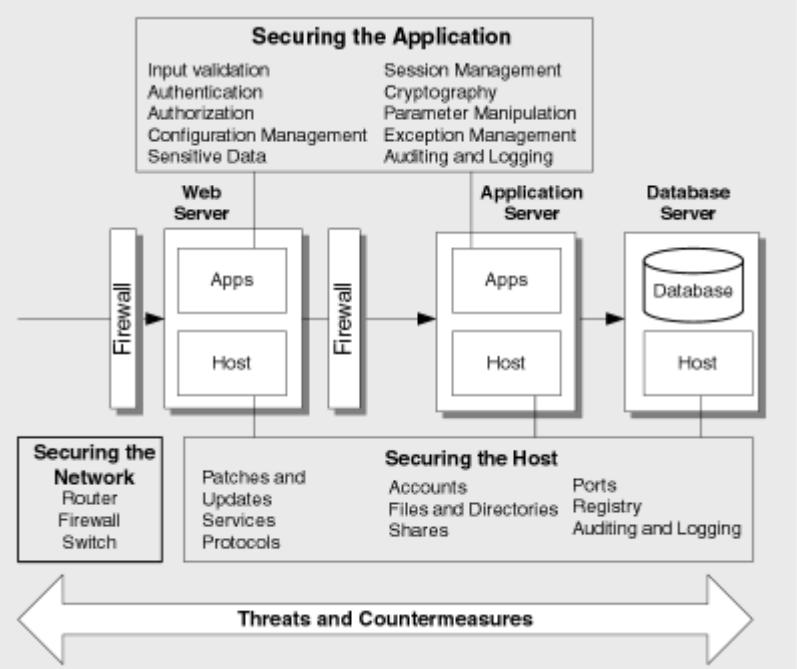

Figura 3 - Schema per la sicurezza dell'applicazione

In questo capitolo vengono descritte alcune raccomandazioni di programmazione che, indipendentemente dall'implementazione di un'applicazione web a tre livelli, Java o .NET, possano essere di supporto ad una programmazione ed utilizzo sicuro delle applicazioni.
Per comprendere pienamente l'obiettivo di ciascuna raccomandazione e renderla più comprensibile, i paragrafi che seguono sono articolati in una breve descrizione delle diverse tipologie di vulnerabilità possibili, sintetizzate nella tabella in figura, con indicazione delle azioni da mettere in atto come contromisura.

Vulnerability Category	Potential Problem Due to Bad Design
Input Validation	Attacks performed by embedding malicious strings in query strings, form fields, cookies, and HTTP headers. These include command execution, cross-site scripting (XSS), SQL injection, and buffer overflow attacks.
Authentication	Identity spoofing, password cracking, elevation of privileges, and unauthorized access.
Authorization	Access to confidential or restricted data, tampering, and execution of unauthorized operations.
Configuration Management	Unauthorized access to administration interfaces, ability to update configuration data, and unauthorized access to user accounts and account profiles.
Sensitive Data	Confidential information disclosure and data tampering.
Session Management	Capture of session identifiers resulting in session hijacking and identity spoofing.
Cryptography	Access to confidential data or account credentials, or both.
Parameter Manipulation	Path traversal attacks, command execution, and bypass of access control mechanisms among others, leading to information disclosure, elevation of privileges, and denial of service.
Exception Management	Denial of service and disclosure of sensitive system level details.
Auditing and Logging	Failure to spot the signs of intrusion, inability to prove a user's actions, and difficulties in problem diagnosis.

Figura 4 – Categorie di vulnerabilità

4.1 Autenticazione

L'autenticazione (Authentication) è il processo volto a determinare se un utente o un'entità sia chi dichiara di essere. In un'applicazione web è facile confondere l'autenticazione con la gestione delle sessioni.
Gli utenti sono tipicamente autenticati tramite userid e password o strumenti simili. Quando un utente è autenticato, viene posto nel browser utente un token di sessione (o cookie). Questo meccanismo permette al browser di inviare il token ogni qualvolta invia una richiesta effettuando, quindi, un'autenticazione di entità.
L'autenticazione di un utente avviene una volta per sessione mentre l'autenticazione d'entità avviene con ogni richiesta.

Userid e password costituiscono nell'ambito del MEF lo strumento più utilizzato per l'autenticazione. Per garantire un adeguato livello di sicurezza è necessario che alle password

siano applicate politiche che ne aumentino la robustezza e che esistano strumenti che permettano la disattivazione di un'utenza qualora non sia utilizzata.
A questo scopo in ambito MEF è stata adottata un'infrastruttura di Access Management e Single-Sign-On per applicazioni Web multilivello di impostazione molto moderna, che presuppone un modello di accesso alle applicazioni basato sul Ruolo ("RBAC", Role Based Access Control). Per un dettaglio maggiore, consultare i documenti di linee guida, inseriti nel raccoglitore degli standard aziendali Consip paragrafo "Standard di programmazione", relativi alle modalità d'integrazione delle applicazioni Java e DOT.NET con Oracle Login Server.
Secondo questo paradigma le applicazioni non gestiscono più in alcun modo le informazioni di anagrafica utenti, che vengono portate fuori dai rispettivi database e gestite in un unico repository LDAP centralizzato.
Al momento dell'autenticazione, l'Access Manager fornisce alle applicazioni la Userid e le credenziali dell'utente che chiede di accedere (ovvero il ruolo per quella applicazione). Alle applicazioni rimane solo la gestione dell'associazione tra le credenziali utente e le funzioni o le viste sui dati che quel particolare ruolo ha diritto/facoltà di esercitare.

L'SSO Server (Single Sign-on Server della Oracle) è il prodotto adottato in ambito MEF e costituisce la naturale estensione del prodotto "Login Server" adottato nel corso del 2001.
Il prodotto SSO mantiene tutte le funzionalità del Login server introducendo la nuova e potente funzionalità di registrazione diretta dei listener HTTP.
I più importanti vantaggi sono:

1. l'integrazione con le applicazioni diventa ancora più immediata, in quanto l'interazione con l'Access Manager è mediata dal Web Server; l'applicazione non deve più verificare continuamente che l'utente sia autorizzato (nella vecchia modalità di integrazione con il sistema di SSO tutte le maschere devono verificare l'esistenza e la validità del cookie di sessione, nella nuova questo è un compito del Web Server). Il processo di autenticazione si svolge nei seguenti passi:
    - installazione di un "plug-in" sul Web Server (attualmente disponibile per Apache e IIS)
    - registrazione del listener sul sistema di SSO
    - configurazione delle URL da "proteggere" (ad esempio : http://applicazioneX/*)
    - quando arriva una richiesta di accesso per una URL protetta, il plug-in la intercetta, verifica che il client richiedente sia autenticato (in caso contrario esegue la redirect sull'Access Manager per il riconoscimento) e quindi concede l'accesso impostando l'informazione nella variabile di enviroment del browser http_sso_user
    - l'applicazione tramite il metodo get_cgi_env acquisisce l'informazione relativa all'utente

- con questa informazione, via protocollo LDAP o via API, l'applicazione reperisce sull'OID le informazioni sul profilo dell'utente.

2. Diviene possibile "proteggere" l'accesso diretto a URL sul Web Server e quindi a risorse fisiche.

Per le interazioni con l'access manager sono stati sviluppati due package, uno per le applicazioni in tecnologia Java (j-sso_sdk), uno per quelle in tecnologia Microsoft (ms_sso_sdk), contententi tutte le API necessarie.

4.1.1 Linee guida per l'integrazione delle applicazioni con SSO

Le applicazioni, per l'integrazione con il sistema di Single Sign On, utilizzeranno come metodo preferenziale la registrazione del web server anche se sarà mantenuto il metodo precedente sia per compatibilità con le applicazioni esistenti sia per gestire l'assenza di plug-in specifici.
Il plug-in necessario per la registrazione del Web Server è oggi disponibile per:

- Apache
- IIS
- Iplanet

Il sistema è quindi compatibile con tutte le applicazioni Oracle (comprese quelle Java) e con quelle su server Microsoft. Le applicazioni già esistenti potranno continuare ad utilizzare il Login Server senza problemi, prevedendo eventuali interventi di manutenzione evolutiva qualora si decida di sostituire i gruppi attuali per la profilazione degli utenti con gli objectclass di SSO. Ma non si ritiene necessario chiedere di stravolgere l'impianto per registrare il Web Server.

L'interazione delle applicazioni con l'SSO dovrà avvenire esclusivamente via API secondo lo schema rappresentato nella figura seguente:

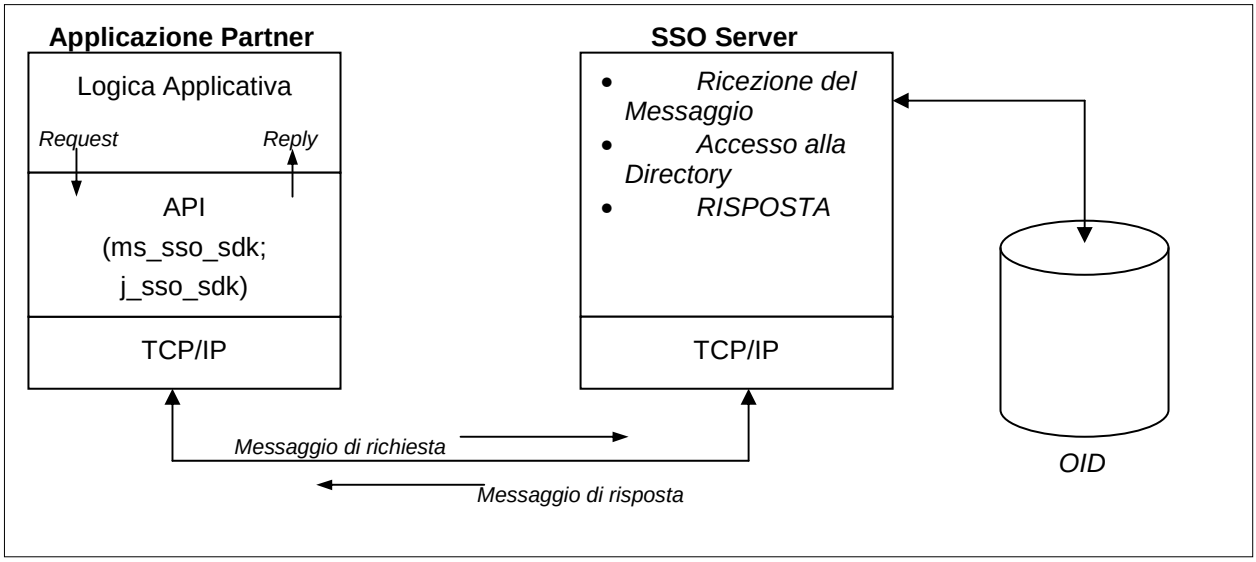

Figura 1 - L'interazione tra applicazioni e SSO Server

L'accesso diretto via LDAP all'OID, pur essendo in linea teorica possibile, è sconsigliato per due motivi, uno di ordine prestazionale, il secondo, più importante, per non "cablare" la struttura DIT dell'LDAP nel codice dell'applicazione (ad esempio, per ottenere via LDAP i valori degli attributi contenuti in un objectclass di un utente è necessario conoscere la collocazione della Entry dell'utente nell'alberatura).

4.2 Controllo dell'accesso ed autorizzazione

I meccanismi di controllo degli accessi (Authorization) costituiscono un elemento cruciale nel disegno della sicurezza delle applicazioni. In generale un'applicazione web dovrebbe proteggere i dati ed i sistemi di front-end e back-end implementando restrizioni su cosa un utente può fare, a quali risorse può accedere e quali funzioni eseguire sui dati.

I termini "autorizzazione" e "controllo degli accessi" sono spesso confusi. Autorizzazione è la verifica che l'utente abbia gli opportuni permessi per accedere a dati e funzioni, l'autorizzazione è legata alle credenziali dell'utente e determina l'appartenenza dello stesso a specifici gruppi preimpostati. E' evidente come qualsiasi controllo degli accessi sia dipendente dal sistema di autenticazione che definisce le autorizzazioni.

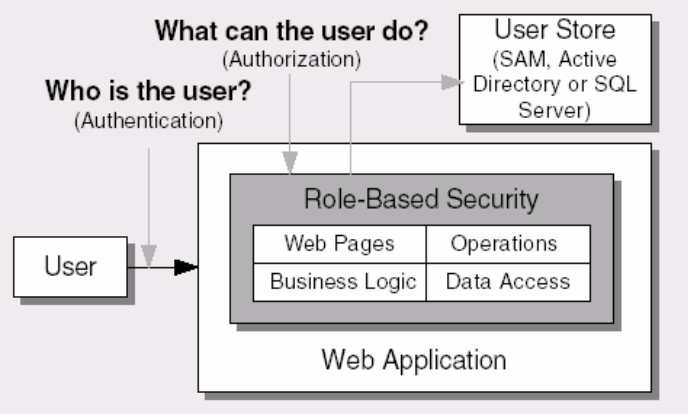

Figura 5 - Autenticazione e Autorizzazione

Il termine "controllo degli accessi" si riferisce più genericamente al modo con cui si controlla l'accesso alle risorse web includendo, tra questi, restrizioni basate su cose quali: l'ora del giorno, l'indirizzo IP di provenienza, il dominio http dell'utente, il possesso di un qualsiasi tipo di token hardware/software.
Esistono diversi modelli/tecnologie di controllo degli accessi ma quelli principali sono:

- DAC (Discretionary Access Control)
- MAC (Mandatory Access Control)
- RBAC (Role Based Access Control)

Il concetto base del DAC è che chi possiede i dati è in grado di controllare l'accesso agli stessi. Le ACL (Access Control List) possono essere considerate un'implementazione del DAC. In questo modello l'accesso alle risorse è basato sull'identità dell'utente e su regole che specificano quale utente abbia accesso a quali dati.

Il modello MAC invece rende sicuri i dati associando a ciascuno di essi una "etichetta" che ne descrive il livello di sicurezza (per es. pubblico, riservato, segreto etc.) e verificando questa "etichetta" con il livello di sicurezza in cui sta operando l'utente. Per determinare se un utente abbia o meno l'accesso a determinati dati, sono utilizzati due principi:

- l'utente può leggere documenti con livello di sicurezza inferiore
- l'utente può scrivere documenti con un livello di sicurezza superiore.

Nel modello RBAC il controllo degli accessi è basato sul ruolo e responsabilità dell'utente nell'ambito di un'organizzazione. Questo modello è stato disegnato per gestire centralmente i permessi definendo dei livelli di astrazione (ruoli) che sono mappati one-to-any con gli utenti,

le funzioni e le risorse. Tale approccio riduce la complessità, in questo modello i permessi sono associati ai ruoli e gli utenti sono associati ai ruoli acquisendo quindi i permessi legati ad essi. I ruoli sono creati e gestiti centralmente e questo rende possibile riassegnare un utente da un ruolo all'altro.

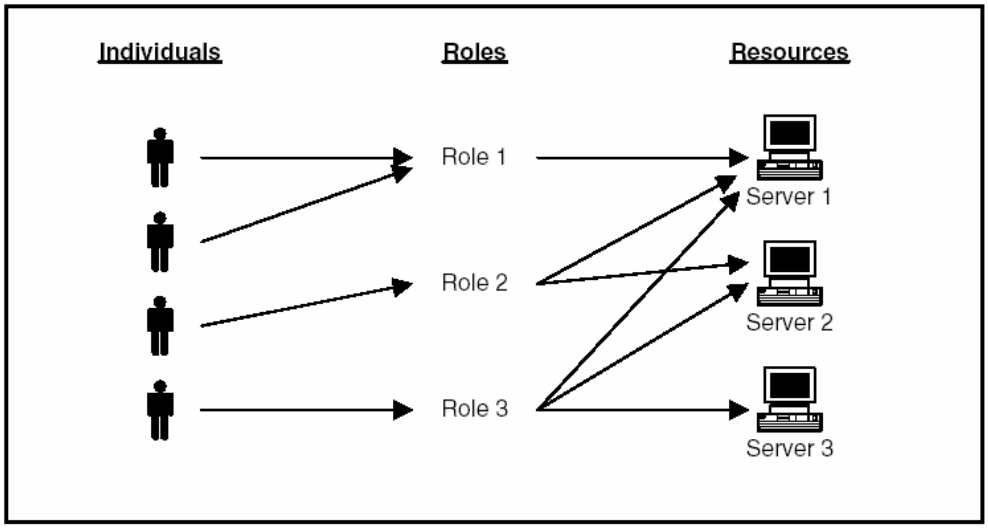

Figura 6 – Modello RBAC

Il modello RBAC è il modello che viene utilizzato, come detto, dalle applicazioni web del MEF. Questa scelta deriva sia dalla facilità gestionale ma soprattutto dalla robustezza insita nel modello. L'utente avrà attribuito solo il livello minimo di permessi necessari per compiere le sue attività, e questo si traduce nell'impossibilità di accedere a dati e funzioni al di fuori dell'autorizzazione assegnata.

4.3 Gestione delle sessioni utente

L'HTTP è il protocollo basato su TCP, utilizzato nelle comunicazioni tra client e server in ambito web. Questo protocollo definisce una semplice interazione di tipo request-response che convenzionalmente è definita "transazione web". L'HTTP è definito come protocollo "stateless" poiché non include il concetto di sessione o interazione che superi la risposta alla richiesta formulata. È quindi necessario, applicare un "meccanismo di stato" affinchè tutte le richieste, provenienti da un unico utente, siano associate tra loro in una "sessione" ("Session Management").

Questo meccanismo è realizzato a livello di applicazione ed una sua implementazione non corretta può determinare severi rischi per l'applicazione stessa.

La maggior parte delle implementazioni di questo meccanismo si basa sull'utilizzo dei "cookies". I "cookies" furono introdotti da Netscape ed oggi sono descritti nel RFC2965. I "cookies" non sono stati disegnati per contenere alcun tipo d'informazione sensibile, questo, insieme ad alcuni accorgimenti, può essere utile per comprendere come utilizzarli

correttamente. I cookies tipicamente contengono un token, ossia un numero unico, non predicibile, in grado di resistere ad operazioni di "reverse engineering", che rappresenta la sessione[1].
Tra gli accorgimenti che devono essere rispettati nel caso di utilizzo di cookies i principali sono:

- Session Time-out: i cookies devono prevedere una scadenza, per limitare le possibilità di riuso, spesso alcuni sistemi prevedono la rigenerazione automatica dopo un numero definito di iterazioni
- Ri-autenticazione: prima di eseguire operazioni critiche è opportuno chiedere all'utente di fornire la propria password
- Trasmissione dei cookies: i cookies devono essere trasferiti esclusivamente su canale crittografato
- Logout: qundo un utente esegue il logoff devono essere cancellati in modo definitivo tutti i cookie.

Si sottolinea che opzioni quali "Remember me" sono assolutamente da evitare.

4.4 Registrazione degli eventi (Logging)

La registrazione degli eventi (logging) è essenziale per fornire elementi chiave relativi ad un'applicazione ed alle operazioni ad essa associate.
Questo log è destinato a registrare principalmente le operazioni di modifica alle basi dati e quelle relativi agli errori che, intercettati dall'applicazione, devono qui essere riportati per poter intervenire a loro correzione.
Questi log:

- possono rivelarsi come uno strumento per individuare comportamenti sospetti
- forniscono una traccia delle azioni degli utenti
- permettono di ricostruire gli eventi dopo che un problema si è verificato, facilitando il processo di recovery
- possono, in alcuni casi, essere utilizzati in processi legali come testimonianza di utilizzo doloso, si noti che in questo ultimo caso l'infrastruttura deve garantire l'inalterabilità dei dati di log.

Gli eventi devono essere registrati accompagnati dall'indicazione della data ed ora dell'evento e dell'utente o processo che ha effettuato l'operazione. Si suggerisce un'ipotesi di nomenclatura quale:

[1] Per implementare ulteriori livelli di sicurezza potrebbe essere necessario vincolare un token di sessione ad una specifica istanza di un client http; per fare ciò un metodo è la generazione di token di pagina e mantenere questa associazione sul server.

XXX_Logging.*

Dove "XXX" è l'acronimo dell'applicazione e "*" l 'estensione del file dipendente dalla piattaforma.

Questi log, da includere nelle procedure giornaliere di backup, devono essere periodicamente (almeno settimanalmente) chiusi, archiviati e ricreati.
Questi log accompagnati dai dati di Audit dei sistemi operativi, DB etc. permettono una completa traccia delle attività in un sistema informativo.
Si ricorda che la scrittura di un file sequenziale è un'operazione onerosa e che quindi tale file non deve essere utilizzato, specie in ambiente di esercizio, per "trace" dettagliate dell'attività dell'applicazione.

4.5 Gestione degli errori

Una non corretta gestione degli errori determina rischi potenziali per un'applicazione. Infatti, senza un'opportuna gestione delle informazioni quali, struttura dei database, versione del sistema operativo, stack etc, possono essere visualizzati all'utente finale.
Questi sono dettagli che, di fatto, disturbano gli utenti e che possono trasmettere, ad un utente malevolo, informazioni utili ad individuare potenziali vulnerabilità.

È quindi necessario applicare alcune best practices nella gestione degli errori:

- presentare all'utente esclusivamente messaggi di errore chiari e comprensibili e comunque non contenenti informazioni o diagnostici non utili a chi legge il messaggio
- registrare dettagliate informazioni di errore su log su filesystem
- inserire codice per la gestione e la cattura di eccezioni non previste. Quest'ultima indicazione fa sì che l'applicazione non rimanga mai in uno stato inconsistente e che l'evento sia registrato sul log per una successiva analisi.

Si rammenta, inoltre, che in ambiente di produzione/esercizio devono essere disattivate tutte le opzioni relative a funzionalità di DEBUG.

4.6 Validazione dei dati di input

La validazione dei dati di input è una delle contromisure più efficaci nella prevenzione di attacchi quali Cross-Site-Scripting, SQL injection, buffer overflows e altri attacchi basati sulla manipolazione dei dati di input.
Per creare una efficace strategia di validazione dell'input esistono tre approcci:

- accettare solo i dati validi
- rifiutare i dati malevoli
- bonificare i dati

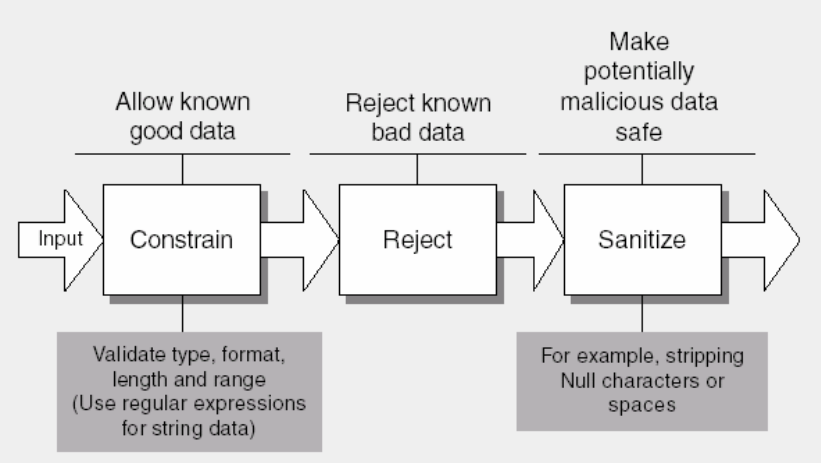

Figura 7 – Approcci per la validazione dei dati di input

Il primo approccio è quello da privilegiare: le applicazioni devono filtrare ogni input basandosi su caratteristiche quali tipo di dato, lunghezza, formato, caratteri ammissibili etc. A queste verifiche realizzate con script dal lato client è però necessario integrare anche verifiche dal lato server. Infatti se la validazione dei dati lato client è accettabile in termini di rapidità e facilità d'uso, di contro, non può essere considerata esaustiva perché esiste la possibilità di disattivare gli script di controllo lato client. È quindi necessario, per una sicurezza più effettiva, che sul server siano presenti routine di validazione dell'input anche se apparentemente questo può essere ritenuto ridondante.

L'approccio che rifiuta dati malevoli richiede che l'applicazione sia in grado di riconoscere i "pattern" /variazione di tali dati. È un approccio meno efficace e soprattutto meno robusto. Infatti mentre i dati validi rimangono costanti nel tempo questo non è vero per i "pattern" dei dati malevoli che cambiano costantemente.

La bonifica dei dati è l'ultimo approccio e prevede di rendere inoffensivi dati potenzialmente dannosi. È un approccio che si rivela utile quando il range dei dati permessi non garantisce che questi siano inoffensivi. Un esempio può essere l'assunzione che nessun input sia considerato come eseguibile ma trattato come semplice testo.

È comunque possibile sommare i tre approcci ma risulta imprescindibile che il primo metodo sia sempre applicato accompagnato da una verifica lato server.

Si sottolinea che tutte le pagine che trattano dati dinamici non devono essere poste in cache e devono prevedere le opportune istruzioni perché siano scartate dai meccanismi di caching.

4.7 Crittografia

4.7.1 Cifratura dati su HTTP

Nei frequenti casi in cui è necessario proteggere integrità e riservatezza dei dati trasferiti, è necessario prevedere l'utilizzo di HTTPS (SSL/TLS su HTTP). Devono essere evitate situazioni in cui vi sia trasferimento d'informazioni fra client e server con protocolli diversi da HTTP.
Per ragioni prestazionali si raccomanda, nel disegno dell'applicazione, di separare, per quanto possibile i dati sensibili da quelli pubblici in modo da limitare l'impiego del protocollo HTTPS alle sole pagine con dati sensibili.

Qualora questo fosse assolutamente necessario, l'invio d'informazioni critiche deve essere comunque protetto o mediante l'applicazione di SSL/TLS ad altri protocolli, oppure mediante altri meccanismi di protezione (es. SSH o simili).
L'invio di password non dinamiche deve essere sempre protetto mediante cifratura.

4.7.2 Meccanismi di non ripudio mediante PIN

Per alcune operazioni è previsto che l'utente debba fornire espressa conferma all'input di informazioni e/o all'attivazione di processi elaborativi o di trasferimento delle medesime. Il meccanismo previsto per impedire che dette operazioni possano essere eseguite senza una effettiva volontà da parte dell'operatore consiste nel codificare il blocco del tasto "Invio". Lo sblocco di detto tasto, e di conseguenza la possibilità di effettuare l'operazione, avviene solo se l'utente inserisce un corretto PIN (PIN dispositivo).

4.7.3 Firma digitale

Secondo gli standard consolidati e la normativa vigente, la firma digitale di informazioni (firma qualificata avanzata realizzata secondo la normativa Italiana precedente al D. Lgs. 10/2002) è realizzata con strumenti esterni alle applicazioni che si interfacciano con l'infrastruttura a chiave pubblica e l'utente e producono documenti firmati in forma di file.

Per quanto riguarda le applicazioni di particolare criticità, deve essere eseguito un controllo sulla corrispondenza del Codice Fiscale indicato nel certificato utilizzato per la firma e quello memorizzato nel database utenti relativo all'utente che ha eseguito l'operazione di firma ed invio delle informazioni firmate.
Se l'applicazione deve gestire informazioni con associata una firma digitale, è necessario, in fase di progettazione e realizzazione dell'applicazione, prevedere quanto segue:

Immissione informazioni firmate

1. L'applicazione deve riconoscere di trovarsi in uno stato in cui è previsto l'input di informazioni firmate ed evidenziare ciò all'utente. L'applicazione deve permettere

all'utente di indicare la locazione del file firmato e di avviare il trasferimento del file mediante un meccanismo di FTP o FTP su HTTP

2. L'applicazione deve procedere alla verifica del formato del file immesso in modo da garantire che i file dichiarati ricevibili siano effettivamente firmati digitalmente.

Gestione informazioni firmate

1. La definizione delle modalità di memorizzazione, elaborazione trasferimento delle informazioni firmate deve tenere conto delle caratteristiche dell'informazione firmata. In particolare deve essere sempre mantenuta l'associazione fra l'informazione e la relativa firma.

Verifica della firma

1. L'applicazione deve poter riconoscere uno stato in cui è necessaria o possibile una verifica di firma di informazioni e fornire all'utente evidenza di ciò
2. L'applicazione deve poter trasferire al sistema di verifica esterno all'applicazione l'informazione nel formato previsto dal sistema di verifica.

4.7.4 Firma del codice applicativo

Le applicazioni WEB, vista l'evoluzione e la sempre crescente complessità che raggiungono, richiedono sempre più spesso l'uso di codice residente sul client e quindi di applet e ActiveX [2].
Con il termine 'applet' viene comunemente indicato del codice scaricabile da server a client unitamente al contenuto di una pagina web HTML. Tale "embedded object" Java è però soggetto a particolari condizioni di sicurezza che ne restringono le potenzialità.
Generalmente si dice che un 'applet' gira dentro una 'sandbox', ovvero un contenitore di sicurezza, dall'interno di tale contenitore, l'applet, e' impossibilitata ad accedere a risorse locali del client, ad aprire connessioni di rete che non siano verso lo stesso host dal quale e' stata scaricata, e così via..
Per ovviare a questa serie di limitazioni è comunque possibile far garantire i permessi necessari ad un applet per poter operare in maniera corretta. A tal fine è necessario disporre di un certificato, e quindi di una coppia di chiavi, con cui 'firmare' elettronicamente un archivio in cui risiede l'applet compressa.
Per gli ActiveX, non esistono le stesse restrizioni, ed è quindi molto più pericoloso scaricarli in locale, in quanto il codice potrebbe contenere virus, dialer o comunque possono essere vettori di codice malevolo. Microsoft ha riconosciuto la debolezza di tale tecnologia, per cui consente agli utenti di bloccare il download di ActiveX non firmati dal browser.

Nell'ambito del MEF, per conciliare la necessità di sicurezza con quella di non impedire l'utilizzo di codice residente sul client, si è scelta la strada della firma di tale codice.
Sono stati acquisiti dalla società Verisign due "Digital ID" ossia una coppia di certificati, uno per oggetti Java ed uno per oggetti Microsoft, che permettono di firmare il codice.

[2] Quanto segue è applicabile anche ad altro tipo di codice destinato a risiedere sul client quali driver, macro Office etc.

Quest'operazione garantisce l'utente che scarica il codice di applicazioni MEF sotto due punti di vista:

- che il codice effettivamente proviene da una fonte nota (Consip per il MEF)
- che il software non sia stato alterato dal momento in cui viene firmato.

5. CONTROMISURE DI SICUREZZA

Le componenti strettamente legate allo sviluppo software e per le quali si forniscono le indicazioni di sicurezza sono definite nella tabella seguente.

Componente	Descrizione
Software	Il componente considera i problemi di sicurezza connessi con il software applicativo
Database	Il componente considera i problemi di sicurezza connessi con l'uso dei database
Web Application	Il componente estende l'analisi sulla sicurezza dei sistemi informativi che adottano tecnologia web based
Pubblicazione Sito	Il componente estende l'analisi di sicurezza sui processi esistenti sulle attività di redazione dei contenuti del sito web
Gestione organizzativa	Il componente considera gli aspetti di sicurezza legati alla gestione dell'applicazione.

Per ogni componente sopra descritta, si fornisce di seguito, sotto forma tabellare, un insieme di contromisure atte a garantire e proteggere la Riservatezza, l'Integrità e la Disponibilità (RID) dei dati trattati dal sistema informativo esaminato.

5.1 Componente Software

Controllo	Contromisura
Covert channels e cavalli di troia	Verificare la "bontà" del software accertandosi in particolare di: 1) disponibilità dei codici sorgenti; 2) verifica e tracciamento, ove possibile, delle modifiche ai codici sorgenti.
Validazione dei dati in input	Verificare che il software preveda controlli specifici per la validazione dei dati inseriti, in particolare è necessario che vengano controllati: 1) dati inseriti fuori dai limiti consentiti (out-of-range); 2) uso di caratteri non ammessi nei dati inseriti (es.: alfanumerici in dati solo numerici o viceversa); 3) dati incompleti (es.: tramite l'utilizzo di maschere per l'inserimento che verifichino la consistenza dello stesso); 4) quantità dei dati inseriti. Inoltre è necessario prevedere procedure per: 1) verifica periodica e al variare degli standard utilizzati delle maschere di inserimento e dei valori per il controllo dei dati inseriti; 2) ispezione dei documenti cartacei utilizzati per l'inserimento dei dati; 3) risposta al verificarsi di errori; 4) test per la verifica dei dati inseriti prima di essere processati; 5) definizione dei ruoli e responsabilità del personale

Controllo	Contromisura
	coinvolto nell'inserimento dei dati; 6) creazione di log delle attività coinvolte nel processo di input dei dati.
Controllo dei processi di elaborazione interni	Utilizzare applicativi che prevedano controlli specifici per la verifica della consistenza dell'elaborazione dei dati con particolare riferimento a: 1) uso delle funzionalità/programmi adibiti all'inserimento/cancellazione e modifica dei dati; 2) procedure per l'inibizione del funzionamento dei programmi in presenza di ambiente non integro (a seguito di failure del sistema); 3) presenza di programmi/funzionalità per il recupero dei dati a seguito di un failure di sistema; 5) protezione contro attacchi che usano tecniche di buffer overruns/overflows.
Validazione dei dati al termine dell'elaborazione	Definire procedure per la verifica dei dati in output dal sistema che prevedano almeno: 1) test di conformità dei dati di output; 2) verifiche automatizzate per la riconciliazione dei dati di output in relazione ai dati di input; 3) una chiara indicazione della precisione e della classificazione del dato stesso; 4) procedure per l'attuazione e la risposta ai test di conformità; 5) definizione di ruoli e responsabilità del personale coinvolto nella raccolta dei dati di output; 6) creazione di log delle attività nel processo di validazione degli output dei dati.
Procedure per il controllo dei cambiamenti	Mantenere uno stretto controllo sulle modifiche apportate al sw o all'ambiente applicativo. In particolare: 1) mantenere un registro dei livelli di autorizzazione del personale adibito ad apportare cambiamenti; 2) assicurarsi che i cambiamenti siano richiesti da utenti autorizzati; 3) revisionare le procedure e i controlli di verifica dell'integrità dell'ambiente e del sw applicativo per accertarsi che queste mantengano la loro efficacia anche a seguito di cambiamenti; 4) procedere all'attuazione dei cambiamenti solo dopo approvazione formale di una proposta dettagliata che descriva le modifiche da attuare; 5) assicurarsi che i lavori di cambiamento siano effettuati ostacolando nel minor modo possibile lo svolgimento delle normali attività lavorative; 6) mantenere aggiornata la documentazione del sw, dell'ambiente, delle procedure operative, ecc. a seguito di cambiamenti e archiviare la vecchia documentazione; 7) mantenere un registro delle richieste di cambiamento.
Protezione delle informazioni trasmesse in rete	Utilizzare protocolli di comunicazione che permettano di proteggere le informazioni trasmesse sulla rete tramite l'utilizzo di algoritmi crittografici.

Controllo	Contromisura
Protezione dei dati di autenticazione (trasmissione)	Utilizzare protocolli di comunicazione che permettano di proteggere le credenziali di autenticazione (password, token, ecc.) trasmessi sulla rete tramite l'utilizzo di algoritmi crittografici
Registrazione degli eventi (audit) – Utilizzare sistemi consolidati di autenticazione (es.SSO)	Il software applicativo deve registrare il verificarsi di eventi significativi dal punto di vista della sicurezza in modo da poter determinare sia l'abuso che l'efficacia del software. Gli eventi che devono essere registrati includono: a) logon e logoff e durata dell'accesso dell'utente; b) tentativi di accesso a risorse e dati riusciti e falliti; c) tentativi di accesso a funzioni di gestione utenti (creazione, cancellazione utenti, ecc.); d) tentativi di accesso a funzioni di policy (modifica permission, ecc.) e) numero e durata delle connessioni stabilite con il database; f) avvio e arresto delle funzioni di audit; g) errori del software. La registrazione deve riportare almeno i seguenti dati: a) identità dell'utente che ha scatenato l'evento; b) data e ora dell'evento; c) tipo dell'evento; d) oggetti coinvolti dall'evento (file, applicazioni, ecc.). Conservare i dati relativi agli eventi registrati per un periodo di tempo sufficiente alla loro analisi e/o utilizzazione a fini statistici, come prove da esibire in caso di dispute, come elementi da considerare nell'identificazione di misure migliorative della sicurezza.
Revisione dei risultati dell'audit	Analizzare a seguito di incidenti o malfunzionamenti, e comunque periodicamente ed eventualmente ad orari casuali distribuiti durante le 24 ore, i risultati dell'attività di audit (utilizzando utility di sistema e/o altri strumenti specifici). Notificare tempestivamente all'amministratore del sistema la scoperta di anomalie (ad es. violazioni della politica di accesso). Proteggere i dati e le funzionalità di audit da accessi non autorizzati. Prevedere funzioni di protezione dei file di audit nei confronti di perdite dovute a saturazione di risorse.
Protezione delle informazioni di log	Controllare che le informazioni contenute nei file di log siano protette da manomissioni e accessi non autorizzati e che non ci siano problemi operativi con le logging facilities. In particolare, occorre verificare che non ci siano: 1) alterazioni ai tipi di messaggio che sono stati registrati nel file di log; 2) problemi con i log files che sono stati pubblicati o distrutti; 3) fallimenti dell'operazione di registrazione degli eventi nel file di log o sovrascrizione degli eventi precedentemente registrati, causati da un superamento della capacità media del file di log.
Sincronizzazione degli orologi	Mantenere costantemente sincronizzato con l'UCT (Universal Coordinated Time) l'orologio a cui fa riferimento

Controllo	Contromisura
	il sistema utilizzando un segnale orario fornito da istituti riconosciuti (come ad es. l'istituto Galileo Ferraris) distribuito attraverso canali dedicati quali: linea telefonica, segnali radio provenienti da stazioni terrestri o satellitari (GPS), ecc.
Procedura sicura di logon- Utilizzare sistemi consolidati di autenticazione (es.SSO)	Il software applicativo deve minimizzare l'opportunità di accessi non autorizzati mediante una procedura di logon che preveda di: 1) non consentire l'accesso fino a quando il processo di logon sia stato completato con successo 2) non fornire messaggi di aiuto, durante il processo di logon, che possono aiutare un utente non autorizzato 3) convalidare le informazioni del logon solo al completamento di tutti i dati di input. Se una condizione di errore si presenta, il sistema non deve indicare quale dato è corretto o incorretto 4) limitare il numero di tentativi di logon non riusciti a 3. Dopo il terzo, imporre un intervallo di tempo prima che un nuovo tentavito di logon sia consentito oppure respingere ogni altro tentativo senza una specifica autorizzazione. Registrare i tentativi di logon sia di successo che di insuccesso. Inviare un messaggio di allarme alla console del sistema se il massimo numero di logon è raggiunto 5) limitare il tempo entro il quale la procedura di logon deve ultimarsi. In caso di eccesso, la procedura deve terminare 6) visualizzare le informazioni seguenti in caso di logon di successo: - data e tempo del precedente logon di successo - dettagliare ogni tentativo di logon di insuccesso dall'ultimo logon di successo 7) non visualizzare la password quando viene inserita oppure considerare di nascondere i caratteri della password con simboli 8) non trasmettere la password in chiaro sulla rete.
Identificazione e autenticazione degli utenti- Utilizzare sistemi consolidati di autenticazione (es.SSO)	Il software applicativo deve regolamentare l'accesso secondo le seguenti modalità:1) assegnare a tutti gli utenti un identificativo univoco (UserID) per il solo uso personale e fornire una adeguata tecnica di autenticazione che verifichi l'identità dichiarata dell'utente.
Gestione della password a livello di software applicativo	Il software applicativo deve gestire le password con le seguenti modalità: a) imporre l'uso di user ID e password individuali per sostenere il principio di accountability b) permettere all'utente di selezionare e cambiare la propria password ed includere una procedura efficace che tenga conto di errori di inserimento c) imporre una qualità della password che tenga conto della lunghezza minima di 8 caratteri d) imporre il cambiamento della password ad intervalli regolari: ogni 3 mesi. Le password assegnate ad utenti

Controllo	Contromisura
	privilegiati devono esser cambiate con frequenza mensile e) imporre il cambiamento della password temporanea al primo logon f) non consentire l'uso o il riciclo delle precedenti 5 password g) non mostrare a video la password quando viene inserita e non dare indicazioni sulla sua lunghezza. h) memorizzare le password in file differenti da dove vengono memorizzati i dati di sistema i) memorizzare le password in forma protetta tramite algoritmi di cifratura o funzioni hash j) trasmettere le password in forma protetta tramite algoritmi di cifratura o funzioni hash
Restrizione dell'accesso alle informazioni	Utilizzare applicazioni che permettono di definire regole di controllo d'accesso con la granularità richiesta dal rispetto del principio del "need to know" (attraverso meccanismi basati su user-id e password e su Access Control List o in assenza di questi, su meccanismi di protezione dei singoli documenti basati esclusivamente su password). Configurare l'applicazione in modo che i risultati delle elaborazioni siano indirizzati solo verso dispositivi di output autorizzati e contengano esclusivamente le informazioni necessarie all'uso a cui sono destinati, verificando periodicamente che essi non includano informazioni superflue. Differenziare la documentazione sulle funzionalità dell'applicazione in funzione della tipologia di utenti a cui la documentazione stessa è destinata.
Presentazione di messaggi di avvertimento– Utilizzare sistemi consolidati di autenticazione (es.SSO)	Includere nella schermata di logon l'avvertimento che l'accesso è consentito ai soli utenti autorizzati. Richiamare le norme interne o di legge che verrebbero violate in caso di accesso non autorizzato e le relative sanzioni. Informare chi si accinge ad accedere al sistema che le attività saranno oggetto di audit.
Mascheramento a video delle chiavi segrete (password, PIN, passphrase) – Utilizzare sistemi consolidati di autenticazione (es.SSO)	Il software applicativo deve gestire le chiavi segrete (password, PIN, passphrase, ecc.) con le seguenti modalità: a) non mostrare a video la password quando viene digitata oppure sostituire i caratteri digitati con simboli; b) non dare indicazioni sulla sua lunghezza.
Controllo accesso alle libreria sorgenti	Manutenere le librerie ed i sorgenti dei programmi rispettando le seguenti linee guida: 1) i sorgenti non dovrebbbero essere conservati in sistemi di produzione; 2) per ogni applicazione dovrebbe essere nominato un responsabile delle librerie di programma; 3) l'accesso del personale dell'IT ai sorgenti dovrebbe essere limitato; 4) programmi in fase di svilupppo o manutenzione non dovrebbero essere mescolati ai sorgenti dei programmi in esercizio; 5) l'aggiornamento delle librerie dei programmi e l'assegnazione ai programmatori di codici sorgenti dovrebbero essere eseguite soltanto dal gestore delle

Controllo	Contromisura
	librerie previa autorizzazione del responsabile IT dell'applicazione; 6) i codici sorgenti dovrebbero essere custoditi in ambienti sicuri; 7) gli accessi ai sorgenti delle librerie dei programmi dovrebbero essere sottoposti ad auditing; 8) le vecchie versioni dei codici sorgenti dovrebbero essere archiviate con indicazione precisa del periodo in cui sono state rilasciate; 9) copia e manutenzione dei sorgenti delle librerie dei programmi dovrebbero essere soggette a rigido controllo.
Protezione dei dati usati durante i test	Ridurre il più possibile l'uso di dati reali per il test di nuove applicazioni. Nei casi in cui ciò non può essere evitato, attenersi alle seguenti regole: 1) depersonalizzare ove possibile i dati di test; 2) applicare le stesse procedure e controlli di accesso previste per i sistemi che trattano i dati reali anche a quelli in fase di test; 3) prevedere che l'uso di dati reali per i test possa avvenire solo dietro esplicita autorizzazione rilasciata caso per caso; 4) cancellare i dati usati per i test, immediatamente dopo la conclusione dei test stessi; 5) mantenere traccia di ogni uso di dati reali in fase di test per consentire operazioni di controllo.

5.2 Componente Database

Controllo	Contromisura
Validazione dei dati in input	Verificare che i dati in ingresso al database siano filtrati da opportuni meccanismi (ad es. stored procedure), in particolare è necessario che vengano controllati: 1) dati inseriti fuori dai limiti consentiti (buffer overflow); 2) uso di caratteri non ammessi nei dati inseriti (ad es. apici, apostrofi, simboli di disuguaglianza, ecc.).
Procedura sicura di logon	Il sistema deve minimizzare l'opportunità di accessi non autorizzati mediante una procedura di logging che preveda di: 1) non consentire l'accesso fino a quando il processo di logon sia stato completato con successo 2) non fornire messaggi di aiuto, durante il processo di logon, che possono aiutare un utente non autorizzato 3) convalidare le informazioni del logon solo al completamento di tutti i dati di input. Se una condizione di errore si presenta, il sistema non deve indicare quale dato è corretto o incorretto 4) limitare il numero di tentativi di logon non riusciti a 3. Dopo il terzo, imporre un intervallo di tempo prima che un nuovo tentavito di logon sia consentito oppure respingere ogni altro tentativo senza una specifica autorizzazione. Registrare i tentativi di logon sia di successo che di insuccesso. Inviare un messaggio di allarme alla console del sistema se il massimo numero di logon è raggiunto 5) limitare il tempo entro il quale la procedura di logon deve ultimarsi. In caso di eccesso, la procedura deve terminare 6) visualizzare le informazioni seguenti in caso di logon di successo: - data e tempo del precedente logon di successo - dettagliare ogni tentativo di logon di insuccesso dall'ultimo logon di successo 7) non visualizzare la password quando viene inserita oppure considerare di nascondera i caratteri della password da simboli 8) non trasmettere la password in chiaro sulla rete.

Controllo	Contromisura
Identificazione e autenticazione degli utenti	Il database system deve regolamentare l'accesso secondo le seguenti linee guida: 1) creare degli account specifici per ciascuna applicazione a cui sono assegnati logicamente una parte o tutti gli oggetti del database, es.: tabelle, indici, procedure, in modo che solo questi speciali ID abbiano i diritti di accesso per modificare tali oggetti 2) creare degli account specifici per tutti gli utenti per il loro uso personale (personale tecnico di supporto, operatori, amministratori e programmatori); 3) agli utenti che eseguono attività ordinarie non devono essere assegnati account privilegiati; 4) Metodi di autenticazione alternativi alla password, come smart card, token, strumenti biometrici e crittografici, devono essere considerati.
Gestione della password a livello di database	Il sistema deve gestire le password con le seguenti modalità: a) imporre l'uso di user ID e password individuali per sostenere il principio di accountability b) permettere all'utente di selezionare e cambiare la propria password ed includere una procedura efficace che tenga conto di errori di inserimento c) imporre una qualità della password che tenga conto della lunghezza minima di 8 caratteri d) imporre il cambiamento della password ad intervalli regolari: ogni 3 mesi. Le password assegnate ad utenti privilegiati devono esser cambiate con frequenza mensile e) imporre il cambiamento della password temporanea al primo logon f) non consentire l'uso o il riciclo delle precedenti 12 password g) non mostrare a video la password quando viene inserita e non dare indicazioni sulla sua lunghezza. h) memorizzare le password in file differenti da dove vengono memorizzati i dati di sistema i) memorizzare le password in forma protetta tramite algoritmi di cifratura o funzioni hash j) trasmettere le password in forma protetta tramite algoritmi di cifratura o funzioni hash
Restrizione dell'accesso alle informazioni	Restringere l'accesso ai dati mediante l'utilizzo di regole di controllo accesso con la granularità richiesta dal rispetto del principio del "need to know", assegnando i diritti di accesso a livello di stored procedure. Differenziare la documentazione sulle funzionalità delle procedure in funzione della tipologia di utenti (es. sviluppatore, amministratore DB) a cui la documentazione stessa è destinata.

Controllo	Contromisura
Limitazione dell'uso delle utility/servizi del database	Limitare l'uso delle utility/servizi del database attraverso: 1) sistemi di identificazione e autenticazione per l'uso delle utility/servizi (restoring o backup del database, ecc.); 2) autorizzazione all'uso delle utility/servizi solo per chi ne ha la reale necessità; 3) tracciamento di ogni operazione svolta con l'uso delle utility/servizi; 4) rimozione di tutte le utility/servizi non strettamente necessarie; 5) disabilitazione di tutte le porte di comunicazione non necessarie presenti sul dispositivo ICT su cui è installato il database; 6) rimozione delle stored procedure non strettamente necessarie e potenzialmente sfruttabili da attaccanti.
Protezione degli strumenti di audit	Controllare l'accesso agli strumenti software di audit in modo da restringerne l'uso ai soli utenti che ne hanno la necessità, conformemente con la politica di sicurezza adottata.
Registrazione degli eventi (audit)	Configurare il database in modo da registrare il verificarsi di eventi significativi dal punto di vista della sicurezza in modo da poter determinare sia l'abuso che l'efficacia del sistema. Gli eventi che devono essere registrati includono: a) logon e logoff e durata dell'accesso dell'utente o applicazione software; b) tentativi di accesso a risorse e dati riusciti e falliti; c) tentativi di accesso a funzioni di gestione utenti (creazione, cancellazione utenti, ecc.); d) tentativi di accesso a funzioni di policy (modifica permission, ecc.) e) numero e durata delle connessioni stabilite con il database; f) avvio e arresto delle funzioni di audit; g) errori del software. La registrazione deve riportare almeno i seguenti dati: a) identità dell'utente o l'identificativo del processo che ha scatenato l'evento; b) data e ora dell'evento; c) tipo dell'evento; d) oggetti coinvolti dall'evento (tabelle, relazioni, dati, ecc.). Conservare i dati relativi agli eventi registrati per un periodo di tempo sufficiente alla loro analisi e/o utilizzazione a fini statistici, come prove da esibire in caso di dispute, come elementi da considerare nell'identificazione di misure migliorative della sicurezza.

Controllo	Contromisura
Revisione dei risultati dell'audit	Analizzare a seguito di incidenti o malfunzionamenti, e comunque quotidianamente ed eventualmente ad orari casuali distribuiti durante le 24 ore, i risultati dell'attività di audit (utilizzando utility di sistema e/o altri strumenti specifici). Notificare tempestivamente all'amministratore del sistema la scoperta di anomalie (ad es. violazioni della politica di accesso). Proteggere i dati e le funzionalità di audit da accessi non autorizzati. Prevedere funzioni di protezione dei file di audit nei confronti di perdite dovute a saturazione di risorse.
Protezione delle informazioni di log	Controllare che le informazioni contenute nei file di log siano protette da manomissioni e accessi non autorizzati e che non ci siano problemi operativi con le logging facilities. In particolare, occorre verificare che non ci siano: 1) alterazioni ai tipi di messaggio che sono stati registrati nel file di log; 2) problemi con i log files che sono stati pubblicati o distrutti; 3) fallimenti dell'operazione di registrazione degli eventi nel file di log o sovrascrizione degli eventi precedentemente registrati, causati da un superamento della capacità media del file di log.
Sincronizzazione degli orologi	Mantenere costantemente sincronizzato con l'UCT (Universal Coordinated Time) l'orologio a cui fa riferimento il sistema utilizzando un segnale orario fornito da istituti riconosciuti (come ad es. l'istituto Galileo Ferraris) distribuito attraverso canali dedicati quali: linea telefonica, segnali radio provenienti da stazioni terrestri o satellitari (GPS), ecc.

5.3 Web Application

Controllo	Contromisura
Conformità agli standard di sicurezza	Eseguire verifiche tecniche per garantire la conformità agli standard relativi alla sicurezza delle applicazioni web secondo i seguenti punti: 1) Effettuare penetration test periodici finalizzati alla verifica della manipolazione di oggetti quali file, directory, database, record, chiavi come URL o parametri di form che possono essere sfruttati da un attaccante 2) effettuare periodicamente la revisione del codice ed eseguire test specifici per verificare che l'autenticazione, la gestione delle sessioni e le funzioni di supporto siano correttamente implementate.
Evitare messaggi dettagliati di errore utili per un attaccante	Eseguire controlli sui messaggi di errore per evitare di fornire informazioni utili ad un attaccante: 1) delineare una policy per la gestione degli errori, inclusi i tipi di errore che devono essere gestiti, quali informazioni devono essere inviate all'utente, e quali informazioni devono essere registrate 2) gli sviluppatori devono comprendere la policy e assicurarsi che il codice prodotto la segua fedelmente 3) sviluppare l'applicazione in modo da poter gestire tutti gli errori possibili. Alcune classi di errore devono essere registrate per semplificare l'individuazione di errori d'implementazione e per tracciare eventuali tentativi di intrusione.
Non eseguire comandi dall'input utente	Eseguire controlli sul codice per assicurare che l'applicazione non esegua comandi presenti nell'input dell'utente: 1) assicurarsi che ogni parte dell'applicazione che permetta un input non processi script come parte dell'input stesso per evitare Cross Site Scripting attack 2) evitare di accedere a risorse esterne e nei limiti del possibile utilizzare librerie specifiche per ogni linguaggio in grado di eseguire la stessa operazione 3) assicurarsi che l'applicazione web venga eseguita solo con i privilegi che necessita assolutamente per eseguire le sue funzioni. In questo modo il webserver non deve essere eseguito come root oppure non si deve accedere al database come DBADMIN 3) se è necessario utilizzare un comando esterno, qualsiasi informazione relativa all'utente che sia inserita all'interno del comando deve essere rigorosamente controllata. Devono essere previsti dei meccanismi in grado di gestire qualsiasi tipo di errore, il timeout o blocchi durante l'esecuzione della call 4) tutti i dati in uscita, i codici di ritorno e quelli di errore della chiamata, dovrebbero essere controllati per assicurarsi che l'elaborazione sia avvenuta.

Controllo	Contromisura
Evitare errori di conversione in formati standard (canonicalization)	Un modo idoneo per effettuare delle conversioni da un formato ad un altro, deve essere concordato dal team di sviluppo in modo che tutti gli input immessi dagli utenti siano validati e trasformati prima di qualsiasi altra operazione. Le verifiche di sicurezza devono essere espletate dopo che la codifica scelta dei dati convertiti (canonicalization) sia stata completata. E' inoltre raccomandato che la codifica scelta sia valida per i simboli che devono essere rappresentati. Ad esempio, controllare che i file eseguibili siano richiamati dalla directory definita a contenerli.
Restrizione sull'accesso a URL	Controllare l'accesso ad ogni URL e ad ogni funzione. Inserire il controllo accessi oltre che nella parte di presentazione anche nella business logic. Il controllo dell'accesso agli URL va realizzato attraverso i seguenti passi: 1) garantire che la matrice del controllo accessi faccia parte dell'architettura e del disegno dell'applicazione 2) garantire che tutti gli URL e le funzioni messe a disposizione siano protette da un efficace meccanismo di controllo che verifichi i privilegi dell'utente prima di effettuare qualsiasi operazione 3) prevenire la possibilità da parte degli utenti di richiamare URL speciali o "nascoste" e garantire che le funzioni e i diritti di amministrazione siano protette 4) bloccare l'accesso a tutti i tipi di file che l'applicazione non dovrebbe mai fornire.
Uso di un session-manager	Impiegare un robusto e ben-consciuto session manager che garantisca un meccanismo di autenticazione di appropriata robustezza. Assicurarsi che il meccanismo non sia facilmente soggetto a spoofing o replay attack e allo stesso tempo non sia eccessivamente complesso. Il web application frameworks deve esser regolarmente aggiornato alla versione più recente in modo da garantire una sicurezza maggiore e l'uso di token crittografici più robusti.
Protezione dei token di sessione	I token di sessione (o session ID) devono essere protetti con SSL.
Rigenerazione dei token di sessione	Il token di sessione deve essere rigenerato: - prima di qualunque transazione significativa - dopo il numero di request stabilite dal responsabile applicativo - dopo 20 minuti di utilizzo per diminuire la finestra temporale della validità del token e ridurre il rischio di session hijacking e brute force attacks.
Time-out della sessione	Il sistema deve automaticamente terminare una sessione dopo 10 minuti di inattività

Controllo	Contromisura
Cancellazione dei dati di sessione al logoff o time-out	Tutti i dati relativi alla sessione, i cookie sul client e lo stato della sessione sul server, devono essere cancellati quando avviene il logoff dell'utente o quando la sessione termina per inattività.
Controllo delle sessioni concorrenti	Il sistema deve limitare il numero di sessioni concorrenti di un utente concordemente con quanto definito dal responsabile applicativo.
Evitare messaggi dettagliati di errore utili per un attaccante	Eseguire controlli sui messaggi di errore per evitare di fornire informazioni utili ad un attaccante: 1) delineare una policy per la gestione degli errori, inclusi i tipi di errore che devono essere gestiti, quali informazioni devono essere inviate all'utente, e quali informazioni devono essere registrate. 2) gli sviluppatori devono comprendere la policy e assicurarsi che il codice prodotto la segua fedelmente. 3) Sviluppare l'applicazione in modo da poter gestire tutti gli errori possibili. Alcune classi di errore devono essere registrate per semplificare l'individuazione di errori d'implementazione e per tracciare eventuali tentativi di intrusione.

5.4 Pubblicazione Sito

Controllo	Contromisura
Separazione dei ruoli	Distribuire i ruoli critici per la sicurezza tra più persone in modo che una singola persona non possa compiere azioni fraudolente senza la collusione di altre oppure commettere abusi e usi impropri con conseguente compromissione dell'informazione. Nella distinzione dei ruoli assicurare che nessuna persona possa accedere, modificare o usare un asset, sia di tipo informativo che infrastrutturale, senza autorizzazione o senza la possibilità di poter scoprire azioni fraudolenti, abusi o usi impropri.
Approvazione alla pubblicazione dell'informazione	Un formale processo di approvazione deve essere eseguito prima che le informazioni vengano pubblicate e rese pubblicamente disponibili. Definire procedure formali per la pubblicazione di informazioni su siti web che prevedano: 1) controllo dell'attendibilità delle fonti da cui sono state tratte le informazioni; 2) protezione delle informazioni nel caso di memorizzazione su supporti temporanei precedenti alla pubblicazione; 3) workflow formalizzato per l'approvazione e pubblicazione delle pagine web; 4) verifica periodica dell'attendibilità delle pagine pubblicate.
Pubblicazione di informazioni in conformità con la legislazione in materia di privacy	Le informazioni rese di pubblico dominio devono essere attentamente controllate in modo tale che le informazioni siano diffuse in conformità con la legislazione e regolamentazione in materia di privacy e protezione dei dati.

5.5 Gestione Organizzativa

Controllo	Contromisura
Creazione di copie di riserva delle informazioni (backup)	Definire una politica di backup che, tenendo conto delle esigenze specifiche nel rispetto delle linee guida a livello aziendale, precisi: 1) il tipo di backup più idoneo (data mirroring, full data, incremental, differential); 2) numero di generazioni di backup che devono essere conservate; 3) tipo di supporti da utilizzare; 4) responsabili del backup; 5) siti e requisiti di conservazione dei backup. La politica di backup dovrà imporre come minimo di: 1) produrre una copia di backup di tutto il software prodotto in proprio; 2) produrre una copia di backup di tutti i dati di sistema almeno una volta al mese; 3) produrre una copia di backup di tutti i dati applicativi almeno una volta alla settimana conservando tre generazioni successive dei backup; 4) produrre una copia di backup dei dati relativi ai protocolli di comunicazione almeno una volta al mese conservando tre generazioni successive dei backup; 5) conservare le copie di backup e la descrizione delle procedure per la loro utilizzazione in un sito distante dal sito principale in modo da evitare che queste possano andare distrutte a seguito di eventi disastrosi che interessano il sito principale; 6) proteggere fisicamente le copie di backup e verificarne periodicamente l'integrità; 7) verificare periodicamente che le procedure di recupero garantiscano il ripristino dell'operatività nei tempi previsti; 8) cifrare il backup dei dati riservati.
Verifica della conformità alla politica di sicurezza generale e agli standard	Verificare con cadenza almeno semestrale, e comunque ogniqualvolta la politica di sicurezza generale subisca modifiche, la conformità di tutte le politiche di sicurezza dei sistemi e delle procedure operative a quanto stabilito dalla politica di sicurezza generale, agli standard e a qualsiasi requisito di sicurezza. Nel caso si rilevino non conformità la direzione dell'organizzazione deve capirne le cause, valutare le azioni da intraprendere e implementarle.
Responsabilità nel momento di cessazione del rapporto di lavoro	Definire le responsabilità anche legali di dipendenti, fornitori o terze parti nel caso di cessazione o cambio di lavoro al momento della stipula dei contratti. La funzione HR con il Responsabile della persona che lascia l'attività o cambia mansione dovrebbe gestire gli aspetti rilevanti per la sicurezza presenti nella procedura. Nel caso di fornitori e terze parti coinvolti la procedura può essere seguita dall'agenzia o società a cui fanno riferimento.

Controllo	Contromisura
Definizione di ruoli e responsabilità	Documentare le mansioni di ciascun responsabile locale della sicurezza precisando i beni e processi di sua competenza, nonché le mansioni e responsabilità di ogni altra figura professionale coinvolta nel trattamento dell'informazione.
Analisi e specifica dei requisiti di sicurezza	Includere nei documenti descrittivi dei requisiti funzionali di nuovi sistemi, o di sistemi in fase di ampliamento, gli aspetti relativi alla sicurezza a livello sia tecnico che procedurale e tenerne conto in fase di sviluppo di software applicativo custom o di selezione di prodotti COTS. Considerare la possibilità di utilizzare prodotti la cui sicurezza è stata valutata e certificata.
Definizione dei criteri di accettazione dei sistemi informativi	Stabilire criteri di accettazione per i nuovi sistemi, per gli aggiornamenti di quelli già esistenti o per il passaggio a nuove versioni di SW. Documentare e sottoporre a test i criteri stabiliti. Condurre test di verifica di conformità a criteri prima dell'accettazione dei sistemi. I suddetti criteri devono tenere conto di: 1) prestazioni richieste al sistema; 2) efficacia delle procedure di recupero da situazioni di errore e di riavvio dei sistemi; 3) conformità delle procedure operative, e dei test a cui sono state sottoposte, agli standard predefiniti; 4) effettiva attuazione delle misure di sicurezza concordate; 5) efficacia delle misure di business continuity; 6) prove che il nuovo sistema non influenzi il regolare funzionamento di quelli esistenti anche nei momenti di maggior carico (ad esempio a fine mese); 7) prove che il nuovo sistema non abbia effetti negativa sulla sicurezza complessiva dell'organizzazione; 8) efficacia dell'addestramento all'uso dei nuovi sistemi. Consultare gli utilizzatori finali del sistema in tutti gli stadi del processo di sviluppo per assicurarsi che il progetto proposto risponda alle esigenze operative.
Pianificazione delle prestazioni dei nuovi sistemi	Monitorare costantemente le esigenze dei sistemi in uso (in termini di potenza di elaborazione, capacità di memoria, velocità trasmissiva, ecc.) in modo da disporre di proiezioni affidabili per la progettazione dei nuovi sistemi e l'aggiornamento di quelli esistenti, per potenziarne la disponibilità e l'efficienza. Ciò permette di evitare il ripetersi di errori di progettazione.

Controllo	Contromisura
Separazione degli ambienti di sviluppo, test, produzione	Separare le attività di sviluppo e collaudo di software da quelle di produzione utilizzando ambienti fisicamente e/o logicamente separati. Impedire l'accesso a compilatori, editor ed altre utility di sistema attraverso i sistemi di produzione se non nei casi in cui ciò sia strettamente necessario. Prevedere differenti procedure di logon per i sistemi di produzione e quelli di test e suggerire agli utenti l'utilizzazione di password diverse per le due tipologie di sistemi in modo da ridurre il rischio di errori. Impedire l'accesso indiscriminato agli ambienti di produzione da parte dello staff di sviluppo. Non copiare dati sensibili nell'ambiente di test.
Sicurezza della documentazione di sistema	Definire procedure per la salvaguardia della documentazione di sistema che prevedano: 1) lista di accesso e accesso selezionato e controllato alla documentazione tramite esplicita autorizzazione; 2) annotazione su opportuno registro di prelevamento di documentazione; 3) definizione di password per l'accesso alla documentazione raggiungibile da reti accessibili al pubblico.
Controllo e autorizzazione delle modifiche sul software	Definire procedure che specifichino come eventuali introduzioni di nuovi sistemi e modifiche al software applicativo già esistente debbano seguire un processo formale di documentazione, specifiche, test, controllo qualità e implementazioni gestite. Il processo deve includere un Risk Assessment, un'analisi degli impatti dei cambiamenti, e le specifiche dei controlli di sicurezza necessari. Le procedure devono assicurare che: 1) i cambiamenti siano stati richiesti da entità autorizzate; 2) i cambiamenti non compromettano la sicurezza o l'operatività del sistema; 3) tutto il software, le entità dei database e i dispositivi hardware coinvolti nei cambiamenti siano stati identificati; 4) la documentazione del sistema e le procedure operative vengano aggiornate; 5) la versione del software utilizzato sia chiaramente identificabile; 6) i cambiamenti attuati siano esaustivamente documentati; 7) il test del nuovo software venga condotto in un ambiente separato da quello di sviluppo e produzione.
Limitazione delle modifiche al software	Scoraggiare la pratica di apportare modifiche ai pacchetti di software acquisiti su licenza in considerazione della difficoltà di valutazione dell'impatto che le modifiche potrebbero avere sulla sicurezza e degli oneri connessi con il mantenimento del software modificato.

Controllo	Contromisura
Definizione di procedure di modifica ai sistemi di elaborazione	Definire procedure che specifichino come eventuali modifiche a apparecchiature, software di sistema e/o applicativo, procedure operative, ecc., debbano essere proposte, autorizzate, attuate. Le procedure devono assicurare che: 1) sia definito un processo formale di autorizzazione dei cambiamenti; 2) sia stata condotta un'analisi del potenziale impatto dei cambiamenti che devono essere testati dopo una opportuna pianificazione; 3) tutte le entità interessate dai cambiamenti siano state avvertite; 4) esistano responsabili in grado di annullare i cambiamenti riportando il sistema allo stato in cui si trovava prima delle modifiche nel caso in cui queste non abbiano successo.
Controllo del software operativo	Adottare misure di sicurezza per l'installazione e l'aggiornamento dei file di sistema e del software sui sistemi di produzione che tengano conto dei seguenti criteri: 1) gli aggiornamenti alle librerie nei sistemi di produzione vanno effettuati solo dalla persona designata come responsabile; 2) i sistemi in produzione debbono contenere, per quanto possibile, soltanto codice eseguibile; 3) prima di installare software su sistemi di produzione questo deve superare una fase di test; 4) è opportuno mantenere un registro degli aggiornamenti del software effettuati; 5) è opportuno conservare copia delle versioni precedenti del software prima di effettuare gli aggiornamenti da utilizzare nel caso in cui le nuove versioni creino problemi; 6) la decisione di installare una nuova release di un software deve essere presa dopo un'analisi delle nuove funzionalità di sicurezza che questo offre e degli eventuali problemi di sicurezza da cui può essere affetto; 7) se l'installazione richiede l'accesso al sistema (fisico e/o logico) da parte del fornitore, consentire l'accesso solo dopo formale autorizzazione, limitare l'accesso al minimo indispensabile e monitorare le attività del fornitore; 8) le vecchie versioni del software dovrebbero essere archiviate insieme alle informazioni richieste e i parametri, le procedure, i dettagli di configurazione e il software di controllo per lo stesso tempo previsto per i dati.
Revisione tecnica delle applicazioni dopo i cambiamenti	Definire procedure di revisione dopo l'aggiornamento periodico del sistema operativo o di installazione patch. Le procedure devono coprire: 1) l'analisi dell'impatto sulle applicazioni degli aggiornamenti al sistema operativo; 2) la verifica della disponibilità di risorse umane ed economiche da dedicare alla fase di test necessaria a seguito dell'aggiornamento; 3) l'impatto dell'aggiornamento sul piano di business continuity.

Controllo	Contromisura
Registrazione/Cancellazione utenti	Definire un processo di registrazione/cancellazione degli utenti ai quali deve essere concesso/revocato l'accesso alle risorse informative aziendali. Il processo deve prevedere almeno: 1) l'uso di User ID individuali in modo che gli utenti possano essere resi responsabili delle proprie azioni ; 2) la verifica dell'autorizzazione all'uso delle informazioni e i servizi rilasciata dal "system owner"; 3) la verifica che il livello di accesso richiesto sia in linea con il principio del need to know; 4) la verifica che la concessione dell'accesso non violi il principio di segregazione dei ruoli imposto in azienda; 5) la firma da parte dell'utente di una dichiarazione dalla quale risultino i diritti concessi e le condizioni alle quali l'accesso può avvenire; 6) la gestione di un archivio contenente l'anagrafica utenti diviso per servizi/applicazioni/dati; 7) l'obbligo di rimuovere immediatamente i diritti di accesso degli utenti che cambiano mansione o dimissionari; 8) la verifica periodica (almeno semestrale) dell'assenza di account inconsistenti, ridondanti o obsoleti e la loro eliminazione.
Selezione e gestione delle password	Regolamentare il processo di allocazione delle password prevedendo almeno che: 1) gli utenti firmino una dichiarazione con la quale si impegnano a mantenere riservate le password individuali e a condividere solo con i membri del gruppo quelle di 'work group'; 2) sia richiesto agli utenti di modificare immediatamente le password temporanee (uniche e non facilmente individuabili) assegnate inizialmente o nel caso in cui un utente abbia dimenticato la propria password; 3) vengano utilizzati mezzi sicuri per la comunicazione delle password iniziali agli utenti e sia loro richiesto di confermare la ricezione delle password; 4) le password non vengano registrate in chiaro su nessun tipo di supporto.

Controllo	Contromisura
Regolamentazione dell'uso delle password	Regolamentare l'uso e la selezione delle password (e dove possibile far rispettare le regole in modo automatico sui sistemi) richiedendo agli utenti di: 1) selezionare password di almeno di 8 caratteri, facili da ricordare; 2) non utilizzare password basate su informazioni personali note (nomi, cognomi, date di nascita, numero di telefono dell'utente o di famigliari, ecc.); 3) non utilizzare password composte da caratteri uguali consecutivi o da caratteri esclusivamente numerici o alfabetici; 4) cambiare la password almeno mensilmente senza utilizzare le precedenti 15 password; 5) mantenere riservata la password (non scriverla su nessun supporto accessibile a terzi quali file sul sistema comprese macro o programmi, supporti cartacei, agende, ecc.); 6) cambiare la password quando vi è il sospetto di compromissione della stessa o del sistema; 7) non condividere la password con altri utenti; 8) non utilizzare la stessa password per l'accesso a sistemi o servizi esterni all'azienda (ad esempio per accedere ad Internet attraverso un account personale ed un proprio ISP).