consip

Linee guida per il piano di test relativo alla sicurezza delle applicazioni web

Versione 1.2

23 Gennaio 2008

TABELLA DELLE VERSIONI

Data	Versione	Descrizione delle modifiche	Cap. /Sez. modificati
Marzo 2006	1.0	Nascita del documento	Tutti
Aprile 2007	1.1	Eliminazione espliciti riferimenti a struttura organizzativa Consip Eliminazione paragrafo Acronimi	1.1
Gennaio 2008	1.2	Inserite indicazioni per allineamento con nuovo standard "Piano di test"	1.1

INDICE

1 INTRODUZIONE

1.1 PREMESSA

La seguente nota tecnica costituisce il naturale completamento del documento "*Linee Guida di Programmazione - Sicurezza delle applicazioni web*", configurandosi come uno strumento operativo a supporto del personale che vada ad effettuare le verifiche di sicurezza per un'applicazione web.
Queste verifiche dovrebbero essere di ausilio per indirizzare le vulnerabilità più comuni di un'applicazione evitando di esporre l'esercizio delle stesse a rischi inaccettabili.

La sicurezza di un'applicazione, come efficacemente rappresentato nella seguente figura, è un processo che coinvolge tutte le fasi dello sviluppo del software (SDLC Software Development Life Cycle). In questo scenario la progettazione dei casi di test, oggetto di questo documento, non ha l'obiettivo di sostituirsi ai *test case* propri di ciascun software/applicazione, ma di costituire una verifica, comune a tutte le applicazioni, nella fase di collaudo/preesercizio *Security Test Case* per garantire un'omogeneità di comportamento di tutte le applicazioni rispetto ad alcune vulnerabilità note e comuni nonché ad alcuni requisiti di sicurezza che sono propri del Sistema informativo del MEF.

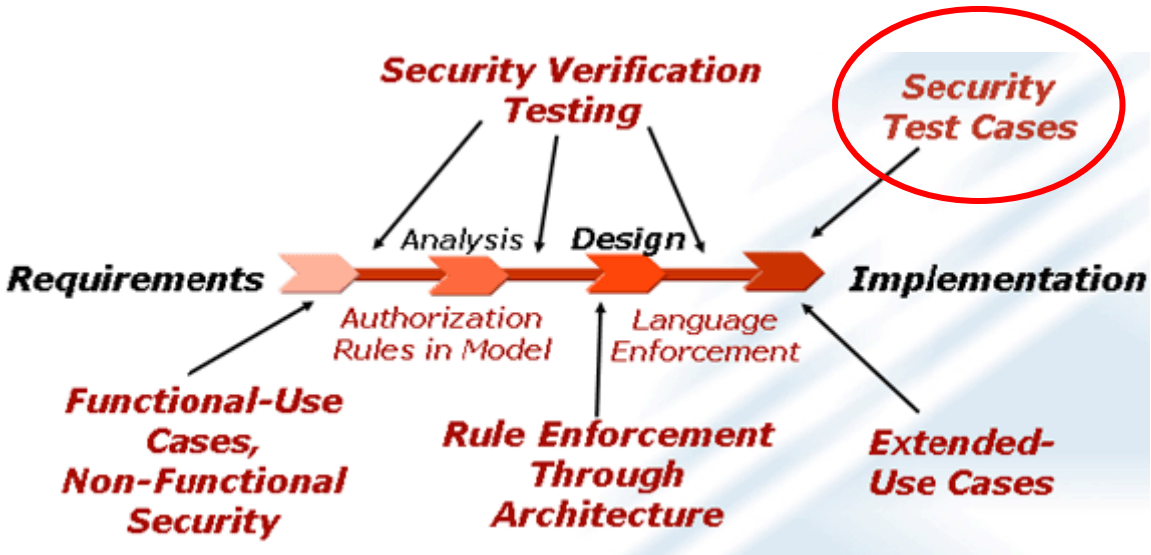

Figura 1

Il documento dà indicazioni sui test che devono essere eseguiti e sui risultati attesi. Le indicazioni, riportate di seguito in una forma tabellare, dovranno essere tenute in considerazione nella progettazione dei test che andrà effettuata nella forma prevista dalla versione corrente dello standard aziendale *"Standard di documentazione - Piano di test"* (al quale si rimanda).

Per ogni controllo viene indicata la modalità di esecuzione. Il superamento di tutti i controlli applicabili è **obbligatorio** per le applicazioni definite critiche secondo il documento "*Linee guida per la definizione della criticità delle applicazioni web e per lo sviluppo di applicazioni critiche*" [1] e per tutte le applicazioni che danno servizio ad utenti esterni. Nel caso di applicazioni non critiche il superamento dei controlli nei casi ambigui verrà valutato congiuntamente dai referenti tecnologici e di sicurezza.

1.2 RIFERIMENTI

[R1] Linee guida di programmazione - Sicurezza delle applicazioni web
[R2] Linee guida per la definizione della criticità delle applicazioni web e per lo sviluppo di applicazioni critiche (*documento ad uso interno Consip*)
[R3] Standard di documentazione - Piano di test

1.3 BIBLIOGRAFIA

Autore	Titolo
Whittaker Thompson	How to break software security
SANS	Varie da: Reading Room
Microsoft	Microsoft Corporation. Design Guidelines for Secure Web Applications

[1] Documento ad uso interno Consip.

2 PROGETTAZIONE DEI CASI DI TEST

2.1 IDENTIFICAZIONE ED AUTENTICAZIONE

2.1.1 VERIFICA DELL'ADEGUATEZZA DEL PROCESSO DI IDENTIFICAZIONE UTENTE O CLIENT

	Verifica dell'adeguatezza del processo di identificazione utente o client
DESCRIZIONE DEL TEST	
I controlli descritti in questa sezione esaminano come gli utenti ed i processi automatici autenticano la loro identità. Perché i controlli siano validi l'ambiente di collaudo deve essere equivalente a quello di esercizio.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Se il processo di identificazione client per l'applicazione è realizzato dal sistema operativo (come nel caso di applicazioni desktop) questo controllo non è applicabile. Per primo è necessario individuare tutti i processi di autenticazione client presenti nell'applicazione. Il termine *client* si riferisce sia a processi utente sia a componenti dell'applicazione (ad esempio un web server si può comportare da client quando si autentica con un database di backend. Anche I processi peer-to-peer sono inclusi in quanto ogni peer agisce indifferentemente come client o come server per alcune transazioni. Ogni processo deve essere valutato separatamente. Se più processi vengono utilizzati per un singolo tentativo di autenticazione, la combinazione dei processi deve essere controllata per garantire che questo controllo sia completamente rispettato.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Per ogni processo deve essere determinata la natura dell'autenticazione, che può coinvolgere uno o più dei seguenti elementi: - Una password (qualcosa che si sa) - Un certificate X.509 (smart card) o un token hardware (qualcosa che si ha) - Un elemento biometrico (qualcosa che si è)

2	In aggiunta, l'autenticazione può coinvolgere l'utenza(account) per il database specifica per l'applicazione o anche un servizio di directory.
3	Se il processo di autenticazione è basato su password, la password deve avere le seguenti caratteristiche: 1. Lunghezza minima di otto caratteri 2. Includere almeno un carattere alfabetico 3. Includere almeno un carattere alfanumerico 4. Includere almeno un carattere speciale 5. Scadere dopo 180 giorni 6. Essere diversa dalle ultime 5 password utilizzate 7. Essere modificabile dall'Amministratore in ogni momento. 8. Essere presente un meccanismo di lockout
4	Verificare la possibilità per l'utente di modificare la propria password
5	Verificare l'esistenza di meccanismi di lookout che impediscano il ripetersi di errati tentativi di logon

RISULTATI ATTESI		
Caso di test		Esito
1	Deve essere presente almeno uno degli elementi richiesti	
2	Per le utenze di acceso a database o directory deve essere verificata la presenza di almeno uno degli elementi richiesti nel caso di test 1	
3	Debbono essere soddisfatte tutte e otto le caratteristiche indicate	
4	L'utente deve essere in grado di modificare la propria password	
5	Il sistema deve bloccare l'accesso dopo n tentativi di accesso con password errata	

2.1.2 VERIFICA CHE L'APPLICAZIONE ACCETTI SOLO CERTIFICATI VALIDI

	Verifica che l'applicazione accetti solo certificati validi
DESCRIZIONE DEL TEST	
I controlli descritti in questa sezione esaminano l'utilizzo dei certificati da parte dell'applicazione. Perché i controlli siano validi l'ambiente di collaudo deve essere equivalente a quello di esercizio.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Se l'applicazione è di tipo web-based ed utilizza certificati lato client deve essere verificata la corretta funzionalità della componente PKI.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Utilizzare una postazione con un certificato revocato; eseguire il logon con il profilo suddetto ed attivare le funzioni dell'applicazione che richiedono l'uso del certificato (per es. l'autenticazione)
2	Utilizzare una postazione con un certificato scaduto; eseguire il logon con il profilo suddetto ed attivare le funzioni dell'applicazione che richiedono l'uso del certificato (per es. l'autenticazione)
3	Utilizzare una postazione con un certificato emesso da una Certification Authority non fidata; eseguire il logon con il profilo suddetto ed attivare le funzioni dell'applicazione che richiedono l'uso del certificato (per es. l'autenticazione)
4	Utilizzare una postazione con un certificato valido; eseguire il logon con il profilo suddetto ed attivare le funzioni dell'applicazione che richiedono l'uso del certificato (per es. l'autenticazione)
5	Eseguire gli stessi controlli per applicazioni non di tipo web.

RISULTATI ATTESI		
Caso di test		Esito
1	Le funzioni non debbono essere attivate. Se una funzione viene attivata segnalare l'anomalia riportando: • quale dei certificati non validi è stato accettato; • da quale delle funzioni testate; • se l' errore è riconducibile ad una anomalia del server.	
2	Le funzioni non debbono essere attivate. Se una funzione viene attivata segnalare l'anomalia riportando: • quale dei certificati non validi è stato accettato; • da quale delle funzioni testate; • se l' errore è riconducibile ad una anomalia del server	
3	Le funzioni non debbono essere attivate. Se una funzione viene attivata segnalare l'anomalia riportando: • quale dei certificati non validi è stato accettato; • da quale delle funzioni testate; • se l' errore è riconducibile ad una anomalia del server	
4	Le funzioni debbono essere attivate. Se una funzione non viene attivata segnalare l'anomalia riportando: • quale dei certificati validi è stato accettato; • da quale delle funzioni testate; • se l'errore è riconducibile ad una anomalia del server	
5	I risultati dovranno essere equivalenti ai quattro test sopra esposti per applicazioni web.	

2.2 GESTIONE DELLE USERID

2.2.1 VERIFICA DELL'UNIVOCITÀ DELLE USERID DELLE APPLICAZIONI

	Verifica dell'univocità delle userid dell'applicazione
DESCRIZIONE DEL TEST	
I controlli in questa sezione analizzano gli account utenti esistenti per valutarne possibili debolezze.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
E' necessario individuare dove le userid sono memorizzate, questo perché alcune applicazioni possono mantenere queste informazioni in posizioni diverse. Se le utenze utilizzate dall'applicazione sono esclusivamente di sistema operativo o utenze di database questi controlli non si applicano.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	É necessario identificare userid duplicate eseguendo un sort delle utenze e ove applicabile utilizzare il codice fiscale dell'utente associato come elemento di riscontro.
RISULTATI ATTESI	
Caso di test	Esito
1	Non deve esistere nessun duplicato

2.2.2 VERIFICA CHE L'APPLICAZIONE CONSENTA ALLA COMPONENTE CLIENT DI AUTENTICARSI ALLA COMPONENTE SERVER CON CUI COMUNICA

	Verifica che l'applicazione consenta alla componente client di autenticarsi alla componente server con cui comunica
DESCRIZIONE DEL TEST	
Questo controllo si applica a tutte le applicazioni che prevedano di autenticare il server con il quale si collegano. L'esempio più comune di tale autenticazione è quando un client autentica il certificato server nel momento in cui instaura una sessione SSL o IPSEC.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Verificare che l'applicazione preveda tale controllo.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Verificare se la funzionalità di autenticazione è stata implementata
2	Per le applicazioni web eseguire il logon per ogni componente che supporta l'autenticazione dei server
3	Eseguire gli stessi controlli per applicazioni non di tipo web.

RISULTATI ATTESI		
Caso di test		Esito
1	La funzionalità deve essere implementata	
2	Nella applicazioni WEB per ogni logon si deve verificare l'autenticazione dei server. Nel caso venga segnalata la non validità del certificato questo può verificarsi in uno o più dei seguenti casi: • Il certificato non è stato emesso da una Certification Authority fidata. • Il certificato è scaduto • Il nome del certificato non coincide con la URL della pagina che si tenta di visualizzare. Se il processo di autenticazione del server non è correttamente innescato dall'azione utente sono da rivedere le configurazioni lato client e lato server. L'autenticazione del server da parte del client deve essere basata su meccanismi crittografici .	
3	I risultati dovranno essere equivalenti ai test sopra esposti per applicazioni web.	

2.3 CONTROLLO DELL'ACCESSO ED AUTORIZZAZIONE

2.3.1 VERIFICA, SE PREVISTO, L'UTILIZZO DIFFERENZIATO DELLE FUNZIONI IN BASE AI RUOLI (MODELLO RBAC)

	Verifica, se previsto, l'utilizzo differenziato delle funzioni in base ai ruoli (modello RBAC)	
DESCRIZIONE DEL TEST		
L'obiettivo di questo controllo è verificare che l'applicazione permetta l'utilizzo delle funzioni ai soli utenti abilitati. In particolare tale utilizzo differenziato è necessario tra: • il personale che rivede e cancella i log di audit ed il personale che esplica altre funzioni; • il personale che crea modifica e cancella le regole di controllo accessi (a dati e funzioni) ed il personale che effettua immissione dei dati o programmazione.		
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)		
Verificare che l'applicazione preveda tale separazione		
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ		
Caso di test		
1	Utilizzare account con i diversi profili/ruoli ed accedere a tutte le funzioni disponibili	
RISULTATI ATTESI		
Caso di test		Esito
1	Ogni account deve poter utilizzare solo le funzioni per cui è abilitato nel rispetto del modello RBAC; nel caso di tentativo di accesso non permesso deve essere segnalato un messaggio di errore.	

2.3.2 VERIFICA SE I PROCESSI DELL'APPLICAZIONE SONO ESEGUITI CON I PRIVILEGI CORRETTI

	Verifica se i processi dell'applicazione sono eseguiti con i privilegi corretti
DESCRIZIONE DEL TEST	
L'obiettivo di questo controllo è verificare che i processi delle applicazioni abbiano i privilegi previsti nei requisiti	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Dai requisiti dell'applicazione si debbono individuare i privilegi necessari ad ogni processo.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Per l'esecuzione di questo controllo è necessario individuare le utenze con le quali i processi dell'applicazioni sono eseguiti. Queste utenze comprendono i processi come definiti da ***Control Panel Services*** (Windows) o ***ps –ef*** (Unix). Per applicazioni a tre livelli devono essere analizzate anche le utenze che collegano un servizio (web server o appplication server) ad un altro (database server). É quindi necessario individuare i gruppi di appartenenza delle utenze e per ciascun gruppo i privilegi associati. È quindi indispensabile verificare che questi utenti non abbiano la proprietà o permessi a files e a directory esterni all'ambito dell'applicazione e per quelli a supporto dell'applicazione abbiano solo i requisiti minimi.

RISULTATI ATTESI		
Caso di test		Esito
1	Deve essere verificata la corretta attribuzione dei privilegi e che i vari gruppi di appartenenza delle utenze li utilizzino correttamente. Ad es. se i privilegi sono eccessivi questa è una violazione. In particolare se l'utenza è membro del gruppo Administrator (Windows) o ha UID 0 (equivalente a root in UNIX) la violazione è grave. Si tratta di violazione grave anche quando l'utenza sia membro del gruppo SYSAdmin (SQL Server) o SYS e System per (Oracle) o possiede privilegi per la creazione cancellazione o modifica di DDL (Data Definition Language).	

2.4 PROTEZIONE DEI DATI

2.4.1 VERIFICA CHE LE CREDENZIALI DI AUTENTICAZIONE O DATI SENSIBILI NON SONO MEMORIZZATI NEL CODICE

	Verifica che le credenziali di autenticazione o dati sensibili non sono memorizzati nel codice
DESCRIZIONE DEL TEST	
.Il codice non deve contenere credenziali di autenticazione o dati sensibili	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
I seguenti controlli sono relativi all'utilizzo di permessi e di strumenti crittografici per la protezione dei dati sia in fase di trasmissione sia in memorizzazione (data at rest). Questi controlli dovranno essere realizzati in ambiente di produzione per escludere il rischio di segnalazioni derivanti da configurazioni non allineate tra l'ambiente di produzione e quello di collaudo.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	É necessario analizzare il codice sorgente, gli script e le form HTML per verficare i casi in cui una password, un certificato o un dato sensibile siano presenti nel codice.
RISULTATI ATTESI	
Caso di test	Esito
1	Nel codice sorgente, negli script e nelle form HTML non debbono mai essere presenti una password, un certificato o un dato sensibile.

2.4.2 VERIFICA CHE RIFERIMENTI A RISORSE DI RETE NON SONO MEMORIZZATI NEL CODICE

	Verifica che riferimenti a risorse di rete non sono memorizzati nel codice
DESCRIZIONE DEL TEST	
L'obiettivo di questo controllo è verificare che il codice non contiene riferimenti a risorse di rete (pathname, URL etc).	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	É necessario analizzare il codice sorgente per individuare i prefissi ed i suffissi più comuni degli URL (i.e., "http://", ftp://, ".it", ".com" etc.) e ove possibile condivisioni di tipo NFS o NetBios così come indirizzi IP.
RISULTATI ATTESI	
Caso di test	Esito
1	Il codice non deve contenere questi riferimenti che invece dovrebbero essere tutti contenuti in file esterni di configurazione.

2.4.3 VERIFICA SE I DATI SENSIBILI DI UN APPLICAZIONE NON SONO ADEGUATAMENTE PROTETTI IN FASE DI MEMORIZZAZIONE

	Verifica se i dati sensibili di un applicazione non sono adeguatamente protetti in fase di memorizzazione
DESCRIZIONE DEL TEST	
I dati di una applicazione devono essere protetti da appropriati permessi sui file che li contengono.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Per i dati dell'applicazione, la necessità di protezione/cifratura viene identificata in fase di analisi di rischio o da specifici requisiti utente. In base alla documentazione è necessario individuare dove siano memorizzati (file dati o database) e come siano protetti. È fondamentale individuare tutti i casi in cui l'accesso sia concesso a "*Everyone*" .	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Solo gli Amministratori ed i processi che costituiscono l'applicazione, per quanto riguarda le informazioni relative a Identificazione ed Autenticazione, devono avere l'accesso ai file contenenti i dati. Il controllo dovrà riguardare anche i backup di questi dati.
2	Nessun altro utente deve avere la possibilità di leggere o di modificare le credenziali di accesso (password)
3	Le credenziali di accesso quali le password devono essere cifrate. Se la cifratura non è attivata e quindi il dato è memorizzato in ASCII o altro formato leggibile è necessario verificare che non sia in chiaro.
4	Per quanto riguarda le chiavi di cifratura nessun processo deve poter scrivere sul file che le contiene. In caso di necessità di aggiornamento o sostituzione i permessi dovranno essere modificati temporaneamente.
5	Verificare che, quando queste chiavi vengono lette, questo avvenga nel contesto (ambito/perimetro) di sicurezza di un account utente o di un processo dell'applicazione (che agisce per conto di un utente) e che l'accesso il lettura sia concesso al solo/i account utente che devono conoscere la chiave per eseguire le funzioni applicative.

RISULTATI ATTESI		
Caso di test		Esito
1	Solo gli Amministratori ed i processi che costituiscono l'applicazione, per quanto riguarda le informazioni relative a Identificazione ed Autenticazione, hanno l'accesso ai file contenenti i dati. Il controllo riguardare anche i backup di questi dati.	
2	Nessun altro utente ha la possibilità di leggere o di modificare le credenziali di accesso (password)	
3	Le credenziali di accesso quali le password sono cifrate oppure se la cifratura non è attivata, e quindi il dato è memorizzato in ASCII o altro formato leggibile, non è in chiaro.	
4	Per quanto riguarda le chiavi di cifratura nessun processo scrive sul file che le contiene. In caso di necessità di aggiornamento o sostituzione i permessi sono modificati temporaneamente.	
5	Quando le chiavi vengono lette, ciò avviene nel contesto (ambito/perimetro) di sicurezza di un account utente o di un processo dell'applicazione (che agisce per conto di un utente); l'accesso il lettura è concesso al solo/i account utenete che conosce la chiave per eseguire le funzioni applicative.	

2.4.4 VERIFICA SE I DATI SENSIBILI DI UN APPLICAZIONE NON SONO ADEGUATAMENTE PROTETTI IN FASE DI TRASMISSIONE

	Verifica se i dati sensibili di un applicazione non sono adeguatamente protetti in fase di trasmissione
DESCRIZIONE DEL TEST	
Per questo controllo i dati si dividono in due gruppi. Dati di Identificazione ed Autenticazione (I&A) e non (I&A). Tutti i dati di I&A devono essere cifrati prima di essere inviati ad esclusione ovviamente degli accessi pubblici..	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Analizzare i requisiti di protezione	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Identificare quale protocollo di autenticazione viene utilizzato e quale, se presente, invia le password in chiaro (per es.., Telnet, FTP and la basic authentication HTTP). In questo caso è necessario verificare che le funzioni di cifratura non siano fornite da un protocollo di più basso livello quali (IPSEC, L2TP, PPTP).
2	Verificare la cifratura in fase di trasmissione tra i layer fisici che supportano l'applicazione. Questo controllo non è necessario nel caso in cui application server e db server risiedano nello stesso server..
3	Per i dati non I&A in base alla documentazione esistente si devono identificare i requisiti di protezione e, se presente un requisito di cifratura, individuare come l'applicazione lo realizzi (funzioni software o di sistema operativo) e quale protocollo utilizzi (IPSEC,SSl. SSH etc.).
RISULTATI ATTESI	
Caso di test	Esito
1	Esiste il protocollo di autenticazione e gli altri requisiti sono soddisfatti
2	Esiste la cifratura in fase di trasmissione tra i layer fisici che supportano l'applicazione.

3	Per i dati non I&A i requisiti di protezione sono rispondenti e, se presente un requisito di cifratura, l'applicazione lo realizza (funzioni software o di sistema operativo) e viene utilizzato un protocollo corretto.	

2.5 GESTIONE DELLE SESSIONI UTENTE

2.5.1 VERIFICA DELL'ESISTENZA DI LIMITI DI SESSIONE PER L'APPLICAZIONE

	Verifica dell'esistenza di limiti di sessione per l'applicazione	
DESCRIZIONE DEL TEST		
Per ogni tipo di sessione che coinvolga utenti o processi è necessario individuarne i limiti.		
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)		
Questa verifica deve essere condotta con il responsabile dello sviluppo dell'applicazione.		
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ		
Caso di test		
1	Individuare i limiti per il numero di sessioni per ciascun utente o processo e il tempo massimo di una sessione *idle* . La verifica può essere condotta o verificando i parametri di configurazione o analizzando il codice sorgente.	
RISULTATI ATTESI		
Caso di test		Esito
1	I limiti esistono sul numero aggregato delle sessioni o sulla durata di una sessione; questo è accettabile in quanto comunque si definiscono dei limiti all'attività di un utente o di un processo. In caso di violazione è necessario verificare se questi limiti esistano e siano imposti nell'ambiente di funzionamento di un'applicazione (per es. il timeout di una sessione TCP impostato a livello di trasporto).	

2.5.2 L'APPLICAZIONE NON DEVE MEMORIZZARE LE CREDENZIALI UTENTE SUL COMPUTER CLIENT DOPO LA CONCLUSIONE DI UNA SESSIONE

	L'applicazione non deve memorizzare le credenziali utente sul computer client dopo la conclusione di una sessione
DESCRIZIONE DEL TEST	
L'obiettivo di questo controllo è verificare che l'applicazione non memorizzi credenziali utente sul computer client dopo la conclusione di una sessione	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
I cookie permanenti sono lo strumento primario con cui un'applicazione memorizza i dati di autenticazione per più di una sessione browser. Se l'applicazione è un 'applicazione web è necessario verificare che le impostazioni di Internet Explorer (o altro browser se previsto) siano tali da avvertire l'utente prima di accetare un cookie proveniente da un'applicazione (controllare in laboratorio se l'impostazione attuale è questa.).	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Loggarsi all'applicazione, eseguire operazioni standard, fare logoff e chiudere il browser. Controllare in /Windows/system32/config/systemprofiles, /Windows/profiles/*xyz*/cookies e in and the /documents and settings/*xyz*/cookies directories (dove *xyz* corrisponde al profile utente) se è presente un nuovo cookie ed in caso affermativo aprirlo con Notepad (o equivalenti) cercando dati di identificazione o autenticazione.

RISULTATI ATTESI		
Caso di test		Esito
1	Non debbono essere presenti dati di identificazione o autenticazione. Si tratta di violazione grave anche il caso in cui IE sia configurato per utilizzare funzionalità quali "Remember me" (o funzioni sinìmilari in altri browser).	

2.5.3 L'UTENTE DI UN'APPLICAZIONE DEVE ESPLICITAMENTE TERMINARE UNA SESSIONE (LOGOUT).

	L'utente di un'applicazione deve terminare una sessione esplicitamente (logout).
DESCRIZIONE DEL TEST	
Verificare che l'utente connesso possa fare logoff	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Per questa verifica è necessario loggarsi all'applicazione e poi effettuare il logout. È necessario controllare che tutti i riferimenti alla sessione vengano correttamente cancellati dopo il logout, tentando di accedere a funzioni dell'applicazione (per esempio referenziando direttamente un URL) senza effettuare nuovamente il login ma senza effettuare la chiusura della sessione browser. Verificare, dopo aver eseguito il Logout, che la funzione indietro del browser non riporti in pagine attive dell'applicazione.
RISULTATI ATTESI	
Caso di test	Esito
1	Il logout avviene correttamente e sono correttamente cancellati tutti i riferimenti.

2.6 REGISTRAZIONE DEGLI EVENTI

2.6.1 L'APPLICAZIONE REGISTRA CORRETTAMENTE GLI EVENTI RILEVANTI PER LA SICUREZZA

	L'applicazione registra correttamente gli eventi rilevanti per la sicurezza
DESCRIZIONE DEL TEST	
I seguenti controlli sono relativi alla tracciatura delle transazioni e degli eventi di sistema. Si precisa, per una corretta esecuzione di questo controllo che il meccanismo che esegue l'auditing di un'applicazione può essere una combinazione di registrazioni provenienti da registrazioni del sistema operativo, dai log dei web server etc.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	È necessario esaminare i file di log dell'applicazione. Al minimo devono essere registrati i seguenti eventi: • Startup e shutdown • Autenticazione • Assegnazione di autorizzazioni/permessi • Azioni di utenti "Administrator" (trusted user) • Invocazione di processi • Tentativi falliti di accesso ai dati • Cancellazione dati • Trasferimento dati

1 (segue)	• Modifiche della configurazione dell'applicazione • Tentativi riusciti o falliti di Logon • Lock temporaneo di un utenza causati da un numero eccessivo di tentativi di accesso falliti. • Per gli eventi I&A l'origine della richiesta (per esempio l'indirizzo IP di origine) • Per letture e cancellazioni il nome dell'oggetto scritto o cancellato. E per ciascuno di essi devono essere riportate le seguenti informazioni: • UserID o identificativo del processo che ha causato l'evento • Data e ora dell'evento • Tipo di evento • Esito (successo o fallimento) dell'evento.

RISULTATI ATTESI		
Caso di test		Esito
1	Tutti gli eventi debbono essere registrati. Se uno o più degli eventi citati non è presente nel log è necessario: • verificare che la configurazione del servizio di audit ne preveda la registrazione • eseguire una transazione che provochi l'evento e verificare che sia correttamente registrato • analizzare il codice per identificare le routine di gestione dell'errore Deve essere presente un meccanismo di registrazione degli eventi propri dell'applicazione (per es Log4J).	

2.6.2 I FILE DI AUDIT DI UNA APPLICAZIONE SONO VULNERABILI A CANCELLAZIONI E MODIFICHE NON AUTORIZZATE

	I file di audit di una applicazione sono vulnerabili a cancellazioni e modifiche non autorizzate
DESCRIZIONE DEL TEST	
Per ciascuno dei file che costituisce l'insieme delle registrazioni di un'applicazione devono essere analizzati i permessi.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Per un sistema Windows i permessi NTFS debbono essere System – Full control, Administrators and Application Administrators - Read, and Auditors - Full Control. Analoghe impostazioni devono essere verificate sugli altri sistemi (per es UNIX)
RISULTATI ATTESI	
Caso di test	Esito
1	I permessi debbono essere rispondenti

2.7 GESTIONE DEGLI ERRORI

2.7.1 L'APPLICAZIONE DEVE PREVEDERE LA POSSIBILITÀ DI GESTIRE ERRORI ED ECCEZIONI.

	L'applicazione deve prevedere la possibilità di gestire errori ed eccezioni.
DESCRIZIONE DEL TEST	
Gli errori dell'applicazione ed i messaggi visualizzati all'utente possono rivelare informazioni che potrebbero essere utilizzate in successivi attacchi.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Per forzare i messaggi di errore di una applicazione si possono sfruttare tre proprietà generali dei dati di input: lunghezza, tipo e valori limite. Immettendo valori che siano più lunghi di quanto si aspetta l'applicazione si può generalmente causare un errore. Qualora questo non appaia possono verificarsi tre possibilità: 1) il dato viene troncato internamente; 2) il dato non è abbastanza lungo; 3) l'applicazione non reagisce ad input difformi. Il secondo ed il terzo caso vanno verificati perché può significare che l'applicazione è sensibile ad attacchi di tipo *buffer overflow*. L'applicazione deve essere dotata di propri meccanismi di gestione dell'errore. Se così non fosse e si poggiasse a meccanismi interni di sistema questa è una violazione grave. Si sottolinea che specie per le applicazioni Java l'esposizione dello stack a fronte di un errore oltre a disorientare un utente non esperto può contenere in chiaro informazioni I&A nonché chiamate verso database interni. I messaggi presentati agli utenti non devono includere nomi e tipi di variabili, stringhe SQL etc. Gli errori dovrebbero contenere un codice (significativo ai fini della diagnostica) ed una chiara e sintetica indicazione per l'utente.
RISULTATI ATTESI	
Caso di test	Esito
1	Le condizioni illustrate debbono essere rispettate completamente.

2.8 VALIDAZIONE DEI DATI DI INPUT

2.8.1 VERIFICA DEI DATI DI INPUT PRIMA DELL'ELABORAZIONE DA PARTE DELL'APPLICAZIONE

	Verifica dei dati di input prima dell'elaborazione da parte dell'applicazione
DESCRIZIONE DEL TEST	
Per l'esecuzione di questo controllo è necessario verificare i piani di test dell'applicazione per determinare che questi comprendano test per la verifica di input non validi. Si definiscono come input non validi dati di tipo o formato non permesso, la manipolazione di stringhe di input (query string), comandi SQL, script tag, etc.). Se i piani di test comprendono queste verifiche il controllo sarà eseguito a campione altrimenti dovranno essere predisposti test più diffusi e dettagliati.	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
Per l'esecuzione di questo controllo è necessario verificare i piani di test dell'applicazione.	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Eseguire il logon all'applicazione e immettere dati non validi per tutti i tipi di utente.
2	Per il controllo delle stringhe di input è necessario verificare che modificando i dati presenti l'utente non possa accedere a dati che gli sono preclusi (per es. data la string www.xx-xx.it/page.asp?xyz=113 si può verificare se sostituendo al 113 un altro dato venga visualizzato un risultato diverso).
3	In modo analogo in campi che possano accettare più di 15 caratteri si può provare a digitare uno scrip tag (<script>) analizzando il comportamento dell'applicazione

RISULTATI ATTESI		
Caso di test		Esito
1	I dati non validi non debbono essere accettati	
2	Debbono essere visualizzati solo dati di pertinenza.	
3	L'applicazione deve rispondere in maniera corretta	

2.8.2 L'APPLICAZIONE NON DEVE ESSERE VULNERABILE AI "BUFFER OVERFLOW"

	L'applicazione non deve essere vulnerabile ai "Buffer overflow"
DESCRIZIONE DEL TEST	
L'obiettivo di questo controllo è verificare che l'applicazione non sia vulnerabile ai "Buffer overflow"	
ATTIVITÀ PRELIMINARI E/O PRECONDIZIONI (*)	
CASI DI TEST E RELATIVA SEQUENZA ATTIVITÀ	
Caso di test	
1	Nella predisposizione di questi test è necessario immettere: • Numeri molto grandi con cifre decimali in campi di input definiti numerici. • In campi numerici numeri positivi e negativi • Grandi quantità di dati (almeno 1024K) nei campo testo Se l'applicazione è un'applicazione web che utilizza stringhe di input (query string) il test dovrebbe comprendere l'immissione di almeno 500 caratteri come parametri della stringa.
RISULTATI ATTESI	
Caso di test	Esito
1	Le condizioni di errore riscontrate debbono essere gestite correttamente dall'applicazione.