OWASP Web Application Penetration Checklist

Versione 1.1

14 Luglio 2004

Questo documento è stato tradotto dalla versione originale in italiano da Massimiliano Graziani (securitymaster@mgx.net).

Contenuti

Immagini

Tabelle

Introduzione

L'attività di *Penetration test* non sarà mai una scienza esatta se la si considera solo per la possibilità di avere una lista aggiornata di tutti i buchi dei sistemi conosciuti. L'attività di *security probe* è soltanto una tecnica adatta a verificare il livello di sicurezza di un sistema esposto su web. Eseguire comunque la sola fase di probe non aiuta a comprenderne le vulnerabilità e le relative contromisure, per questo vi consigliamo di leggere la seguente documentazione:

- *OWASP Testing Framework Part One* manuale disponibile su http://www.owasp.org, una buona base per iniziare a comprendere un'architettura complessa su web e le relative problematiche di sicurezza.
- *Risk Management Guide for Information Technology Systems* manuale NIST 800-30 [1], che approfondisce le modalità operative e tecniche di amministrazione.

Cos'è OWASP

OWASP è un'organizzazione *no profit* dedicata alla realizzazione e diffusione di best practice per lo sviluppo su Web, ma anche di documentazione e software di supporto a sistemisti ed architetti di rete, sviluppatori e security manager. OWASP promuove e aiuta gli utenti a costruire un Web più sicuro.

Per maggiori informazioni è disponibile il sito http://www.owasp.org. ed il relativo capitolo italiano: http://www.owasp.org/local/italy.html

[1] http://csrc.nist.gov/publications/nistpubs/index.html#sp800-30 La versione rivista può essere trovata qui: http://csrc.nist.gov/publications/drafts/SP800-30-RevA-draft.pdf

Feedback

Per inviare un vostro feedback relativo al *penetration testing* e alle liste di vulnerabilità, per favore inviate una e-mail a testing@owasp.org con oggetto:

[Pen Testing Checklist Feedback].

Accogliamo favorevolmente qualsiasi osservazione o suggerimento. Se inviate informazioni circa nuovi tipi di vulnerabilità siete pregati di inviare della documentazione sufficiente alla pubblicazione sulle nostre liste. Se invece inviate un'osservazione o un suggerimento siete pregati di inviare un documento completo delle soluzioni proposte. Essendo un'associazione su base volontaria, i vostri suggerimenti ed interventi rappresentano la miglior via per migliorare le nostre revisioni.

Penetration Test Checklist

Molti membri dell'organizzazione OWASP, in particolar modo le aziende di servizi finanziari, hanno chiesto la realizzazione di una lista ufficiale di vulnerabilità stilata, organizzata e gestita dall'OWASP. Tale lista verrà considerata un punto di riferimento per i test di penetrazione. Saranno quindi disponibili:

- Template per Request for Proposal (RFP)
- Benchmark
- Checklist di Test

La checklist fornirà un solo elenco di vulnerabilità suddivise per sistema e revisione, non fornirà invece la prescrizione di tecniche che dovrebbero essere usate.

Utilizzare la Checklist come template RFP

Molti utenti sentono l'esigenza di poter avere un documento di riferimento da poter utilizzare sia per i controlli autonomi, sia per i controlli da richedere a terze parti. Questo perché è necessario avere un punto di riferimento standard per poter applicare degli SLA (Service Level Agreement). E' così che la Checklist può essere considerata come un RFP tra cliente e fornitore. Infatti, chi richiede un servizio di Penetration Test richiede l'applicazione di un test perlomeno sulle vulnerabilità aggiornate, ovvero quelle inserite nella Checklist condivisa da OWASP.

Nota*: Se la vostra azienda sviluppa un Template RFP partendo dalla nostra Checklist, siete pregati di condividerla con OWASP e con tutta la comunità informatica. Inviatela a testing@owasp.org con il seguente oggetto: [Testing Checklist RFP Template].*

Utilizzare la Checklist come Benchmark

Molti utenti hanno la necessità di avere un punto di riferimento per effettuare dei test prestazionali sul *Penetration Test*. Usando una Checklist comune è possibile effettuare dei test comparativi tra diversi sistemi o metodi di *security probe*.

Il progetto OASIS sulla Web Application Security (WAS) (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was) diventerà un punto di riferimento dove saranno classificate le vulnerabilità e le metodologie. Per maggiori informazioni, potete prendere visione del capitolo “Lo Standard OASIS WAS “ presente in questo documento.

Utilizzare la Checklist come Checklist

Naturalmente molte persone vorranno utilizzare la lista semplicemente come controllo per i propri sistemi. La lista comunque non fornisce le tecniche da utilizzare, anche se contempla alcuni esempi.

Informazioni sul Progetto di Test OWASP (Parte Uno e Due)

OWASP sta attualmente lavorando per creare un Testing Framework. Mentre state leggendo questo documento, la *Parte Uno del Testing Framework* verrà completata e la *Parte Due* sarà in lavorazione. La *Parte Uno* descrive il perché, il come, dove e quando dei test sulla sicurezza delle Applicazioni Web. La *Parte Due* riguarderà dettagli tecnici su come cercare codice vulnerabile ed effettuare un penetration test; ad esempio, come cercare un bug di tipo “SQL injection” nel codice sorgente o con un tool di penetration test. Questa checklist potrebbe diventare, insieme alle altre liste di controllo per il codice sorgente, un’appendice della *Parte Due dell’OWASP Testing Framework* .

Lo Standard OASIS WAS

I controlli presenti nella checklist non sono ordinati per importanza o criticità.

Molti utenti dell’OWASP Team stanno lavorando ad uno standard XML per lo sviluppo che sia in grado di classificare le problematiche di sicurezza OASIS per le applicazioni su web. La missione di OASIS è quella di guidare lo sviluppo, la convergenza e l’adozione di informazioni strutturate e standard nell’area dell’e-business, dei web services ecc.

Per maggiori informazioni circa OASIS consultate il sito http://www.oasis-open.org.

Noi crediamo che OASIS WAS diventerà uno standard molto importante che permetterà a molte persone lo sviluppo di una gestione dei rischi e delle vulnerabilità a prescindere della tipologia dei dati trattati. Poiché questo approccio è attualmente un punto di riferimento indipendente da fornitori e tecnologie (per questo la sua longevità è garantita), OASIS WAS rappresenta lo standard su cui basare il vostro lavoro.

Una parte dell'OASIS WAS standard sarà un *set* di tipologie di vulnerabilità. Queste sono vulnerabilità che permetteranno una classificazione in grado di fornire un supporto per lo sviluppo. Utilizzando questa classificazione gli utenti possono creare report standard del proprio stato di sicurezza applicativa.

Lo standard OASIS WAS XL è stato pubblicato nell'Agosto 2004. Le tipologie di vulnerabilità WAS sono state pubblicate in un documento a parte alla fine di Aprile 2004. Comunque, anche se improbabile, questa lista in caso di necessità viene sottoposta periodicamente a revisione.

Noi crediamo che le tipologie di vulnerabilità WAS diventeranno parte integrante della gestione della sicurezza applicativa. Questa affermazione sarà strettamente contemplata in tutte le iniziative e documentazioni prodotte dal gruppo di lavoro OWASP come la Checklist ed il Testing Framework.

Penetration Testing Workflow

Promuovendo tale checklist, proponiamo chiaramente anche una metodologia per effettuare test ripetibili.

Sebbene non è compito primario di questa Checklist fornire una metodologia di *penetration test* (questo sarà contemplato nella *Parte Due dell'OWASP Testing*), un modello di metodologia di test sarà incluso, come indicato in Figura 1. Chi effettua il test troverà utile seguire le tecniche di test descritte in questo documento. E' importante notare come i livelli dell'infrastruttura del *penetration test* lasciano la possibilità di agire sia in modalità superficiale che approfondita. In alcuni casi, i sistemi possono essere esplorati in base ai permessi che l'utente ha ricevuto dall'amministratore del sistema dove risiede l'applicazione web.

Il diagramma a blocchi in Figura 1 si basa sui seguenti passi:

1. Il *penetration test* inizia quando si hanno a disposizione tutte le informazioni possibili circa il sistema da analizzare. Questa fase è obbligatoria; senza informazioni le attività successive non possono essere svolte.
2. Il test deve seguire tutti i passi indicati in Figura 1.
3. I *tester* dovrebbero tentare di sfruttare tutte le vulnerabilità conosciute. Anche se il test non avrà il successo sperato, si acquisirà una maggiore comprensione della materia. Ogni informazione rilevata dal test di vulnerabilità (ad esempio, errori di programmazione, recupero di codice sorgente, e altre informazioni riservate) dovrebbe essere utilizzata per comprendere e migliorare la sicurezza in ambito applicativo.
4. Se in qualsiasi punto del test venisse rilevata una vulnerabilità che crea un disservizio o produce la fuga di informazioni, chi esegue il test deve contattare

immediatamente il proprietario dell'applicazione/sistema ed avvisarlo documentando la vulnerabilità ed indicando le eventuali soluzioni.

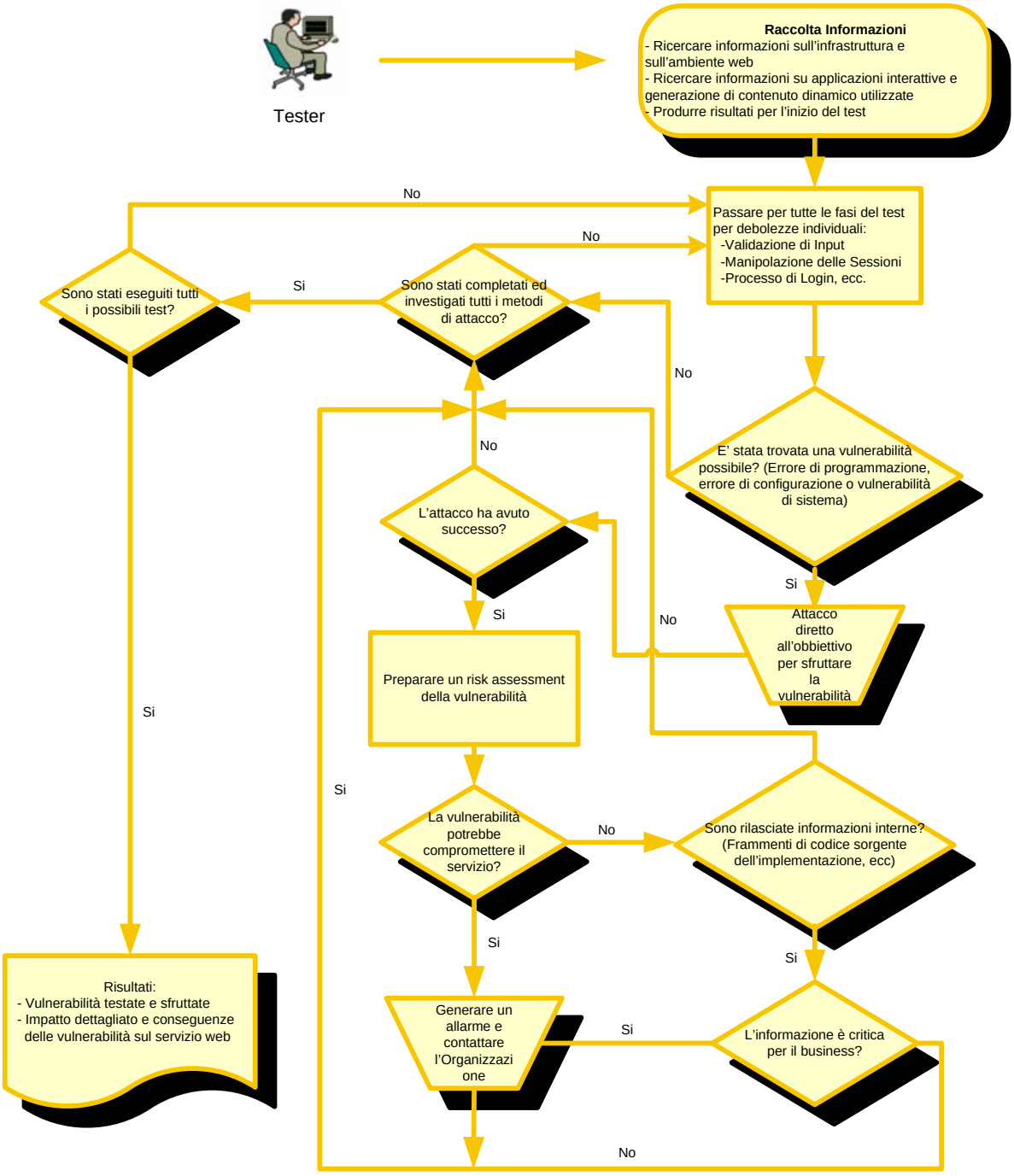

Figura 1: Modello di Metodologia di Test.

Checklist

La Tabella seguente è la *Penetration Testing Checklist* corrente:

Tabella 1: Pen Test Checklist

Categoria	Numero di Rif.	Nome	Obiettivo	Note
DOS Applicativo	OWASP-AD-001	Flooding dell' applicazione	Accertarsi che le funzioni applicative lavorino correttamente anche quando ci sia un enorme volume di accessi e traffico di rete.	Utilizzare diversi *fuzzing tool* per questo tipo di test (come ad esempio, SPIKE)
	OWASP-AD-002	Blocco Applicazione	Accertarsi che un attaccante non riesca a bloccare o a resettare i diritti di accesso degli utenti autorizzati.	
Controllo degli accessi	OWASP-AC-001	Analisi Parametri	Verificare che l'applicazione forzi l'uso del suo modello di controllo di accesso assicurando che ogni parametro disponibile ad un attaccante non colpisca servizi addizionali.	Di solito questi includono la manipolazione dei campi di un *form*, di stringhe di richieste URL, valori di script lato client e nei cookie.
	OWASP-AC-002	Autorizzazione	Assicurarsi che si effettuino controlli di autorizzazione adeguati prima di inviare risorse protette a un utente.	
	OWASP-AC-003	Manipolazione dei Parametri di Autorizzazione	Assicurarsi che una volta che un utente valido si è loggato, non sia possibile cambiare i parametri della sessione per riflettere quelli di un altro account.	Ad esempio: numero di account, numero di policy, numero utente, ecc. Questo permettere un furto di identità.

Categoria	Numero di Rif.	Nome	Obiettivo	Note
	OWASP-AC-004	Pagine/funzioni Autorizzate	Controllare se sia possibile accedere a pagine o funzioni che richiedano un login ma che possa essere bypassato.	
	OWASP-AC-005	Workflow Applicazione	Assicurarsi che, dove l'applicazione richieda all'utente di effettuare delle azioni in una specifica sequenza, questa sia obbligata.	
Autenticazione	OWASP-AUTHN-001	Richieste di Autenticazione tra *end point* dovrebbero essere in HTTPS	Assicurarsi che agli utenti sia richiesto di inviare credenziali di autenticazione solo in pagine fornite in SSL.	Ciò assicura che l'utente sappia chi sta richiedendo le sue credenziali e dove sono inviate.
	OWASP-AUTHN-002	Bypass Autenticazione	Assicurarsi che il processo di autenticazione non possa essere bypassato.	Di solito succede con attacchi come *l'SQL Injection.*
Autenticazione. Utente	OWASP-AUTHN-003	Trasporto delle Credenziali attraverso un canale criptato	Assicurarsi che username e password siano inviati attraverso canali criptati.	Di solito dovrebbe essere in SSL.
	OWASP-AUTHN-004	Account di Default	Controllare che non vi siano nomi e password di *default* in uso.	
	OWASP-AUTHN-005	Nome utente	Assicurarsi che il nome utente non sia un'informazione pubblica come l'email o il SSN.	
	OWASP-AUTHN-006	Qualità della Password	Assicurarsi che la complessità della password rispetti dei parametri di lunghezza e tipologia di caratteri adeguati	

Categoria	Numero di Rif.	Nome	Obiettivo	Note
	OWASP-AUTHN-007	Reimpostazione Password	Assicurarsi che l'utente debba fornire una risposta segreta o una domanda segreta o altre informazioni predefinite prima che la password possa essere reimpostata.	Assicurarsi che le password non siano inviate alle e-mail degli utenti.
	OWASP-AUTHN-008	Blocco Password	Assicurarsi che gli account utenti siano bloccati per un periodo di tempo quando viene inserita una password errata per un numero di volte maggiore di quello specificato (di solito 5).	
	OWASP-AUTHN-009	Struttura Password	Assicurarsi che non si possano usare caratteri speciali all'interno delle password.	Tornano utili nell'effetturare SQL injection.
	OWASP-AUTHN-010	Password Vuote	Assicurarsi che venga inserita una password. non vuota.	
Autenticazione. Gestione Sessioni	OWASP-AUTHSM-001	Lunghezza del Token delle Sessioni	Assicurarsi che il token delle sessioni sia di adeguata lunghezza per fornire protezione dal tentativo di indovinarlo durante una sessione autenticata.	
	OWASP-AUTHSM-002	Scadenza Sessioni	Assicurarsi che i token siano validi solo per un predefinito periodo di tempo dall'ultima richiesta dell'utente.	
	OWASP-AUTHSM-003	Riutilizzo delle Sessioni	Assicurarsi che i token delle sessioni siano cambiati quando un utente si sposta da una risorsa protetta da SSL ad una non protetta.	

Categoria	Numero di Rif.	Nome	Obiettivo	Note
	OWASP-AUTHSM-004	Eliminazione delle Sessioni	Assicurarsi che il token di sessione sia invalidato quando l'utente termina la sessione con l'applicativo.	
	OWASP-AUTHSM-005	Formato del Token delle Sessioni	Assicurarsi che il token di sessione sia non persitente e non sia mai scritto nella history del browser o nella cache.	
Configurazione. Gestione	OWASP-CM-001	Metodi HTTP	Assicurarsi che il server web non abbia la possibilità di manipolare risorse da Internet (PUT e DELETE).	
	OWASP-CM-002	Siti con Hosting Virtuale	Cercare di determinare se il sito sia su un host virtuale.	Se ci sono ulteriori siti potrebbero essere vulnerabili e portare alla compromissione del server di base.
	OWASP-CM-003	Vulnerabilità note/Patch di Sicurezza	Assicurarsi che vulnerabilità note e già patchate dei vendor non siano presenti.	
	OWASP-CM-004	File di Back-up	Assicurarsi che non siano accessibili sulla parte accessibile al pubblico dell'applicazione file di backup del codice sorgente.	
	OWASP-CM-004	Configurazione del Server Web	Assicurarsi che problemi comuni di configurazione come lista delle directory e file di esempio siano stati affrontati ed eliminati.	

Categoria	Numero di Rif.	Nome	Obiettivo	Note
	OWASP-CM-005	Componenti del Server Web	Assicurarsi che le componenti del server web (come le estensioni di Frontpage o i moduli di Apache) non introducano vulnerabilità nella sicurezza.	
	OWASP-CM-006	Percorsi Comuni	Controllare l'esistenza di directori comuni all'interno della root dell'applicazione.	/backup & /admin potrebbero contenere informazioni.
	OWASP-CM-007	Defualt di Linguaggi/Applicazioni	Ad esempio problemi di environment di J2EE; la disponibilità di snoop.jsp/*Spy.jsp e dei moduli caricati.	
Configurazione. Gestione. Infrastruttura	OWASP-CM-008	Interfacce di amministrazione Infrastruttura	Assicurarsi che le interfacce di amministrazione all'infrastruttura, come web server e application server, non siano accessibili da Internet.	
Configurazione. Gestione. Applicazione	OWASP-CM-009	Interfacce di Amministrazione Applicazione	Assicurarsi che le interfacce amministrative dell'applicazione non siano accessibili da Internet.	
Gestione Errori	OWASP-EH-001	Messaggi di Errore Applicazione	Assicurarsi che l'applicazione non presenti messaggi di errore che possano essere usati in un attacco da un attaccante.	Ciò di solito capita quando le applicazioni mostrano messaggi di errore dettagliati come tracce dello stack o errori di database.
	OWASP-EH-002	Messaggi di Errore Utente	Assicurarsi che l'applicazione non presenti messaggi di errore utente che potrebbero essere usati in un attacco a un attaccante.	Ciò di solito succede quando le applicazioni ritornano messaggi del tipo "L'Utente non esiste" o "Utente Corretto, Password errata".

Categoria	Numero di Rif.	Nome	Obiettivo	Note
Protezione Dati	OWASP-DP-001	Dati Sensibili in HTML	Assicurarsi che non ci siano dati sensibili nell'HTML (memorizzato nella history del browser) che possano guidare un attaccante a costruire un attacco preciso.	Ciò accade di solito quando gli sviluppatori lasciano commenti nell'HTML o quando l'applicazione invia nomi e indirizzi in HTML.
	OWASP-DP-002	Memorizzazione Dati	Assicurarsi che i dati siano protetti per assicurare la loro confidenzialità e integrita ove richiesto.	
Protezione Dati. Trasporto	OWASP-DP-003	Versione SSL	Assicurarsi che le versioni di SSL supportatte non abbiano debolezze crittografiche.	Di solito significa supportare solo SSL 3.0 e TLS 1.0.
	OWASP-DP-004	Metodi di scambio chiavi SSL	Assicurarsi che il server web non permetta metodi di scambi chiavi anonimi.	Di solito ADH Anonymous Diffie-Hellman.
	OWASP-DP-005	Algoritmi SSL	Assicurarsi che non siano disponibili Algoritmi SSL deboli.	Di solito algoritmi come RC2 e DES.
	OWASP-DP-006	Lunghezza chiavi SSL	Assicurarsi che il sito web usi una lunghezza di chiave appropriata.	Molti siti web dovrebbero applicare crittografia con chiavi a 128 bit.
	OWASP-DP-007	Validità del Certificato Digitale	Assicurarsi che l'applicazione usi certificati digitali validi.	Assicurarsi della validità del certificato digitale: la sua firma, l'host, la data, etc devono essere validi.
Validazione Input	OWASP-IV-001	Script Injection	Assicurarsi che ogni parte dell'applicazione che permetta un input non processi script come parte del'input stesso.	Classico caso di *Cross Site Scripting*, ma include anche altro scripting.

Categoria	Numero di Rif.	Nome	Obiettivo	Note
Validazione Input. SQL	OWASP-IV-002	SQL Injection	Assicurarsi che l'applicazione non esegua comandi SQL dall'input utente.	
Validazione Input.OS	OWASP-IV-003	OS Command Injection	Assicurarsi che le applicazioni non processino comandi del sistema operativo dall'input utente.	Di solito include problemi del tipo *path traversal*, *spawning command shells*, e funzioni del sistema operativo.
Validazione Input.LDAP	OWASP-IV-004	LDAP Injection	Assicurarsi che l'applicazione non processi comandi LDAP dall'input utente.	
Validazione Input.XSS	OWASP-IV-005	Cross Site Scripting	Assicurarsi che l'applicazione non memorizzi o inoltri codice script dannoso.	

Appendice A - OASIS WAS Tipi di vulnerabilità

ControlloDegliAccessi

Tipo di problema che può consentire a determinati utenti di accedere a risorse o servizi, per i quali non dispongono di autorizzazione. Non di rado capita di non ritrovare alcun meccanismo di controllo degli accessi, dove invece dovrebbe essere implementato. Un meccanismo di controllo degli accessi valido deve rispettare i requisiti tipici di un "*reference monitor*": ovvero dovrebbe essere *tamperproof* e dovrebbe essere verificabile.

DOSApplicativo

Difetti nel software che possono condurre gli utenti a non utilizzare appropriamente un applicazione o a renderla completamente inutilizzabile.

DOSApplicativo.Flood

Utilizzato negli attacchi di tipo "*denial of service*" che prevedono la saturazione di una risorsa condivisa tra più utenti, per esempio: CPU, banda, connessioni ad un database, memoria, ecc.

DOSApplicativo.Blocco

Utilizzato negli attachi di tipo "*denial of service*",che prevedono l'utilizzo di una risorsa o di un limite allocato a un utente, come i tentativi falliti di login, i messaggi o le transazioni.

Autenticazione

Problemi che possono insorgere durante la procedura di identificazione degli utenti o di generiche entità (es: server), e nel processo di autenticazione.

Autenticazione.Entità

Utilizzato per problemi con sistemi automatici di autenticazione, come servizi web, database, directory ed altro. Alcuni esempi includono la memorizzazione sicura di credenziali, la messa in sicurezza del trasporto, il cambio delle credenziali e la chiusura d'accesso.

Autenticazione.GestioneSessioni

Utilizzato per problemi di creazione, d'uso protezione, cambio e terminazione di identificatori di sessioni di tutti i generi. Gli identificatori di sessione rimpiazzano le credenziali di autenticazione, ma sono ancora protetti in maniera adeguata.

Autenticazione.Utente

Utilizzato per problemi relativi all'identificazione ed all'autenticazione di persone che possono usare un'applicazione. Sono degli esempi i problemi con nomi utenti, password, token, smartcard, sensori biometrici, ed altre credenziali.

Autenticazione.GestioneUtenti

Termine usato per riferirsi a problemi collegati alla gestione degli utenti, specificatamente per le informazioni critiche da un punto di vista della sicurezzza, quali ruoli, autorizzazioni, privilegi, gruppi, codice fiscale, numeri di carte di credito, e altre informazioni sensibili; comprende anche le problematiche inerenti la creazione di nuovi utenti, di registrazioni a servizi, assegnazione di privilegi e accesso alle risorse.

BufferOverflow

Bug del software che può permettere ad un attaccante di sovrascrivere spazi di memoria allocata dai processi software, utilizzando ad esempio stringhe opportunamente formattate, e che può portare a modifiche nei dati , nel flusso del programma o anche a crash dell'applicazione.

BufferOverflow.Format

Errore dell'applicazione che può consente ad un attaccante di sovrascrivere spazi di memoria allocata dai processi software, tramite l'utilizzo di stringhe opportunamente formattate, consentendogli di modificare i dati , il flusso del programma o causare un crash dell'applicazione.

BufferOverflow.Heap

Errore del software che può permettere ad un attaccante di causare un *overflow* dallo spazio di memoria riservato per l'allocazione dinamica.

BufferOverflow.Stack

Errore dell'applicazione che può consentire ad un attaccante di scrivere dei dati nello stack di memoria, causando il crash dell'applicazione o il trasferimento del flusso di programma.

Concorrenza

Si presenta a causa di errori di progettazione degli ambienti *multithread*, e può portare alla condivisione non voluta o all'alterazione dei dati. Un caso comune è quello di variabili condivise tra più *thread* che causano problemi *di time-of-check-time-of-use* (TOCTOU) , violazione del *pattern singletone*, e errata gestione della cache.

GestioneConfigurazione

Termine usato quando ci si riferisce ai problemi collegati alle interfacce di configurazione delle applicazioni o degli ambienti applicativi.

GestioneConfigurazione.Amministrazione

Termine usato in riferimento a problemi collegati alle interfacce remote per le funzioni di amministrazione quali: gestione utenti, gestione delle credenziali, gestione dei database, ecc..

GestioneConfigurazione.Applicazione

Termine usato per riferirsi a problematiche relative alla configurazione delle applicazioni, ad esempio errate configurazioni delle funzionalità di sicurezza, , dei programmi di base, del codice non utilizzato e di funzionalità presenti ma non necessarie.

GestioneConfigurazione.Infrastruttura

Termine usato per riferirsi ai problemi connessi alla configurazione dell'infrastruttura applicativa, quali ad esempio webserver, application server, componenti di filtering, e strumenti per la sicurezza

Crittografia

Termine usato per riferirsi a problematiche connesse alla cifratura, decifratura, firma digitale e verifica della firma.

Crittografia.Algoritmo

Termine utilizzato in un contesto di selezione degli algoritmi crittografici ed di questioni di implementazione/analisi degli algoritmi.

Crittografia.GestioneChiavi

Usato in riferimento a questioni inerenti lo *storage* di certificati digitali e delle relative chiavi private, token crittografici, revoca dei certificati, *key storage* ed emissione delle chiavi..

ProtezioneDati

Termine usato per riferirsi ad una non appropriata divulgazione di dati.

ProtezioneDati.Memorizzazione

Termine usato per questioni inerenti il salvataggio in sicurezza di dati, comprendenti la memorizzazione di credenziali, chiavi crittografiche ed altre informazioni sensibili. Malfunzionamenti legati ai meccanismi crittografici consistono in sorgenti con randomicità non sufficiente, scelta errata di algoritmi, e cattiva implementazione.

ProtezioneDati.Trasporto

Termine usato ad indicare problemi nel trasferimento in sicurezza di informazioni. Generalmente si fa riferimento a problemi di configurazione di SSL e TLS, ma può anche riferirsi a generici protocolli di comunicazione aventi funzionalità di sicurezza.

GestioneErrori

Termine usato per riferirsi ad una cattiva gestione degli errori. In particolare la visualizzazione a schermo dello *stack*, fallimenti nell'attivazione di meccanismi di sicurezza o nel consentire che il verificarsi di determinati errori puntuali abbiano ripercusioni sul funzionamento globale dell'applicazione; infine nel divulgare a seguito di una condizione di errore informazioni non necessarie.

ValidazioneDatiInIngresso

Termine usato in contesti in cui viene a fallire il meccanismo di validazione dei messaggi di input provenienti da una fonte non fidata, con successivo processamento degli stessi da parte dell'applicazione a cui sono destinati.

ValidazioneDatiInIngresso.File

Si riferisce ai problemi nella fase di *input validation*, in cui l'input dell'applicazione è costituito da un file: file di configurazione, file batch, tracciati di database su flat-file o altri tipi di dati codificati su file.

ValidazioneDatiInIngresso.Utente

Utilizzato per riferirsi a problemi di *input validation*, ove l'input viene inserito da parte dell'utente: parametri di un richiesta HTTP, input a linea di comando o interazione tramite un interfaccia grafica.

ValidazioneDatiInIngresso.Rete

Termine utilizzato per riferirsi a problemi di *input validation*, in cui l'input proviene dai parametri di un protocollo di rete: *header* HTTP, numeri progressivi di protocollo o altri campi di protocollo.

Injection

Criticità in cui è possibile che un attaccante inserisca dei comandi non leggittimi nascosti all'interno di dati legittimamente destinati ad un sistema, che li eseguirà al ricevimento degli stessi.

Injection.HTML

Vulnerabilità che può permettere ad un attaccante di inserire dell'HTML in un'applicazione e di modificare l'apparenza dell'HTML generato da essa. Per esempio un attaccante potrebbe inserire un tag IMG non voluto in un *guest book*, ed offendere gli altri utenti.

Injection.ComandiAlSistemaOperativo

Vulnerabilità che può permettere ad un attaccante di inserire caratteri speciali e comandi nella *shell* di comando del sistema operativo e di modificarne il comando stesso.
L'attacco può cercare di modificare come un programma viene richiamato o può tentare di concatenare comandi addizionali.

Injection.LDAP

Debolezza che può permettere ad un attaccante di inserire caratteri speciali e termini di ricerca in un server LDAP e modificare la query.

Injection.SQL

Vulnerabilità che può permettere ad un attaccante di inserire caratteri speciali e comandi in un database SQL e di modificarne la query. L'attacco può tentare di cambiare il significato della query o può tentare di concatenare comandi addizionali.

Injection.XSS

Vulnerabilità che può permettere ad un attaccante di inviare ed eseguire script nocivi attraverso applicazioni web. Gli attacchi XSS memorizzati registrano gli script nelle applicazioni web. *Reflected XSS* utilizzano un'applicazione web come "ponte" in tempo reale e richiedono che un'utente invii la richiesta contenente l'attacco.

Monitoring

Utilizzato per problemi relativi al controllo delle *policy* di sicurezza di un'applicazione web.

Monitoring.Logging

Usato per problemi riguardanti il corretto log degli eventi, incluso ciò che dovrebbe essere "loggato", come i log dovrebbero essere rivisti ed altri problemi relativi all'*accounting*.

Monitoring.Detenzione

Usato per problemi relativi al rilevamento di attacchi su di una applicazione, a come gli attacchi dovrebbero essere trattati, le informazioni che dovrebbero essere raccolte, e chi dovrebbe essere notificato.

Indice