INTERVISTA
29 novembre 2010

CONTROLLARE LA SICUREZZA DEL SOFTWARE
Intervista al Prof. Danilo Bruschi

Ordinario presso il Dipartimento di Informatica e Comunicazione dell'Università degli Studi di Milano

La sicurezza applicativa è l'area della sicurezza informatica che controlla e gestisce la sicurezza dei dati sul layer applicativo, contribuendo a sviluppare, consolidare e integrare i controlli che garantiscono, la protezione dei dati a livello di confidenzialità, integrità e disponibilità.
Oggi, il mercato italiano mostra come l'applicazione di metodologie di gestione del ciclo di vita del software sicuro sia ancora in fase embrionale, evidenziando da una parte l'assenza di un approccio organico alla gestione della sicurezza delle informazioni (organizzazione, applicazioni, sistemi, reti e protezione fisica), dall'altra la scarsa integrazione con gli aspetti di IT Governance (processi ed organizzazione della gestione del ciclo di vita dei servizi IT), nonostante le recenti normative nazionali e internazionali stiano facendo aumentare l'interesse sul tema.
Se fino a pochi anni fa si pensava che una rete controllata fosse sufficiente a proteggere i dati in transito, oggi, community di esperti di livello internazionale come OWASP (the Open Web Application Security Project) e WASC (Web Application Security Consortium), sottolineano i vantaggi dell'adozione di una metodologia di sviluppo del software sicuro, tra cui la possibilità di controllare il processo di sviluppo non solo negli aspetti dei costi e dei tempi ma anche sotto il profilo qualitativo e la trasparenza sul rispetto dei requisiti non funzionali da parte dei fornitori.
L'adozione di questo modello introduce controlli sul software relativi ad autenticazione, autorizzazione, controllo accessi e gestione delle sessioni, validazione dei dati in input ed output, crittografia, logging ed error handling, verifica delle configurazioni software e auditing, collocando la sicurezza applicativa nel ciclo di gestione e controllo più vasto dell'IT Governance.

Per una maggiore comprensione dei punti più complessi della materia ci avvarremo delle competenze del Prof. Danilo Bruschi, Ordinario di Informatica presso il Dipartimento di Informatica e Comunicazione dell'Università degli Studi di Milano dove insegna nei corsi di Sistemi Operativi, Sicurezza dei Calcolatori e delle Reti e progettazione di Sistemi di Sicurezza. Il Prof. Bruschi svolge attività di ricerca nei settori della Sicurezza delle Reti dei Calcolatori, della Computer Forensic, della Survivability e della Privacy. É inoltre responsabile del Laboratorio di Sicurezza e Reti (LASER) e dirige dal 1995 il Computer Emergency Response Team Italiano.

Secondo una stima del Gartner il 75% degli attacchi sono rivolti al layer applicativo e l'80% delle aziende potrebbe avere un incidente di sicurezza applicativa entro il 2010. Quali sono gli attacchi con gli impatti più pericolosi diffusi sul software?

In questi ultimi anni, caratterizzati in ambito sw dall'esplosione delle applicazioni web, abbiamo assistito alla nascita di diversi prodotti mirati a facilitare enormemente il lavoro di chi è addetto allo sviluppo di applicazioni web. L'effetto è stato che chiunque si è sentito autorizzato a sviluppare codice web, se da una parte tutto questo ha avuto un'evidente ricaduta sui costi di sviluppo dall'altra la qualità del sw prodotto è notevolmente scaduta. I primi a farne le spese in queste situazioni sono i requisiti di sicurezza.

La stragrande maggioranza delle applicazioni web che analizziamo (al laboratorio LASER, n.d.r) sono vulnerabili. Gli attacchi più pericolosi sono quelli di tipo"injection" (iniezioni di codice malevolo o dati, ndr), ma i pericoli sono tanti e differenti. Ad esempio, un paio di anni fa, su alcune applicazioni web abbiamo trovato una vulnerabilità di tipo "race condition" (accesso concorrente a dati condivisi fra utenti e/o processi, ndr) sfruttabile con diversi attacchi e conseguenze particolarmente importanti.

Oggi Wikileaks ci mostra come una vulnerabilità (in questo caso di tipo organizzativo, ndr) possa compromettere la confidenzialità di documenti riservati.

In ambito corporate gli attacchi spaziano dal carpire informazioni (spionaggio industriale svolto su commessa e condotto da abili professionisti), allo sfruttamento di macchine presenti in azienda come testa di ponte per attacchi verso altri target o come deposito di file ed informazioni per usi illeciti. Gli attacchi sono molto frequenti e, in generale, se esiste una vulnerabilità, la possibilità che questa possa essere sfruttata è molto alta.

In ambito enterprise il processo di sviluppo del software viene controllato attraverso indicatori di misurazione dei costi e dei tempi di produzione, mentre la qualità - quando ciò avviene - viene monitorata tramite indicatori che rilevano il numero di incidenti avvenuti od il grado di soddisfazione degli utenti nell'uso del software. É possibile rendere oggettiva e non subordinata alla percezione degli utenti od al manifestarsi di anomalie, la valutazione della qualità del codice applicativo attraverso indicatori per la misurazione della qualità durante lo sviluppo?

L'informatica è molto in ritardo sull'analisi della qualità anche a causa della natura del software: è stato dimostrato che stabilire la correttezza di un programma è un problema indecidibile, in soldoni questo significa che non esistono e mai potranno esistere strumenti per poter accertare o meno la presenza di software bug.

Lo sviluppo del software, accademicamente parlando, è ancora un'arte e, non a caso, Donald Knuth che aveva avuto questa intuizione, scrisse nel lontano 1962, "The art of computer programming".

Oggi, le nuove metodologie permettono di ridurre il numero di errori presenti sul codice ma siamo ancora lontani dalla possibilità di garantire che il codice non presenti difetti.

La direzione migliore da intraprendere è sicuramente quella di avvalersi di risorse competenti e di framework di sviluppo noti. Tutto questo però è spesso in antitesi con le esigenze di mercato, come la velocità di esecuzione, l'usabilità e l'economicità: se si vuol sviluppare un'applicazione sicura si introducono dei rallentamenti perché si inseriscono dei controlli e se si potesse fare a meno delle password… sarebbero tutti più contenti!

È sempre un'impresa ardua convincere il management aziendale del ROSI (Return On Security Investment - http://rosi.clusit.it/pages/Homepage.html, ndr) se poi questo management si è formato e cresciuto all'ombra della cultura main frame, l'impresa diventa impossibile.

Un approccio che in Italia ha dato qualche segnale positivo è la "sicurezza per legge". Difatti la fase in cui il nostro paese ha prestato più attenzione alle problematiche di sicurezza informatica è stata quella relativa all'attuazione della legge sulla privacy. Anche in questo caso però sarebbe necessario disporre a livello politico di competenze che sappiano introdurre con un adeguato livello di importanza concetti di garanzia e protezione delle informazioni, ma considerato il livello di preparazione tecnologica della nostra classe politica, mi rendo conto che sto parlando di fantascienza.

Esistono diversi standard più o meno consolidati per lo sviluppo applicativo sicuro (come quelli ideati da OWASP e Microsoft). Quali reputa più efficaci?

Come accade in ogni nuovo settore le metodologie evolvono insieme, prendendo spunto anche l'una dall'altra, di conseguenza non devono essere utilizzate in modo esclusivo ma vanno integrate tra loro. OWASP possiede il forte vantaggio del modello open-source di cooperazione a livello internazionale che le permette di evolvere e migliorare rapidamente.

Già da qualche anno si osserva l'abbandono, da parte di un numero sempre crescente di aziende, di processi di produzione del software rigidamente strutturati (a cascata) a favore di metodologie cosiddette "agili". É possibile integrare in maniera analoga al modello di sviluppo a cascata test di sicurezza e controlli?

Dal 2005 sono iniziate a livello internazionale una serie di attività per introdurre i requisiti di security nell'ambito di metodologie basate sull'agile computing piuttosto che sull'extreme computing.

Oggi sia OWASP che Microsoft approfondiscono i concetti di gestione della sicurezza su modelli di sviluppo sia rigidamente strutturati che agili, seppur l'ultima frontiera dell'ingegneria del software sia quella di integrare a livello euristico la sicurezza in fase di compilazione.

Siamo ancora in una fase embrionale che attraverso dei tentativi cerca di ridurre le occasioni di errore di un programmatore. Il problema è che oramai il software ha acquisito un'importanza estremamente elevata e gli utenti si aspettano un'affidabilità pari a quella degli oggetti di uso quotidiano, come un televisore o un frigorifero. Sul software non si riescono ad innestare schemi di standardizzazione e controllo della correttezza, come può avvenire in altri settori dell'ingegneria (elettronica, meccanica…); ma, siamo ottimisti, probabilmente nell'arco di 100-150 anni (sic!), cominceremo ad ottenere quelle migliorie nei processi produttivi che porteranno all'abbattimento delle vulnerabilità.

Quando la decisione di investire in sicurezza non è spinta da obblighi di compliance si tende spesso a sottovalutare o a non valutare affatto, i rischi per il proprio business. É possibile individuare le industry e/o le tipologie di software che dovrebbero valutare necessariamente l'adozione di un Secure SDLC?

Come no! Vi ricordate nel 2000 il Millennium Bug? Il Consiglio dei Ministri aveva istituito un comitato di crisi che aveva adottato una serie di misure precauzionali per prevenire eventuali impatti sulle infrastrutture nazionali ed avrebbe dovuto coordinare le attività di soccorso. Di fatto non successe nulla ma il comitato svolse un lavoro molto valido nell'individuare i settori la cui erogazione di servizi fosse critica per le attività del paese. Questo lavoro può oggi essere attualizzato, individuando attraverso apposite metodologie, le realtà più critiche in termini di interdipendenza. A tutte queste realtà dovrebbe essere imposto uno standard qualitativo da rispettare, perché i danni in caso di incidente si ripercuoterebbero non solo sull'azienda ma su tutti coloro che ne interdipendono.

Le necessità ed opportunità per le aziende di focalizzarsi sul proprio business e quindi di esternalizzare lo sviluppo delle applicazioni sw a terze parti (outsourcer), pone nuovi problemi di controllo e richiede criteri di scelta e valutazione per assicurare predefiniti standard di qualità e sicurezza. Quali sono i requisiti da valutare per stimare l'attendibilità e l'affidabilità di un fornitore di software sicuro e quali sono le certificazioni che permettono di attestare la sua conoscenza della sicurezza applicativa? Le aziende che scelgono l'outsourcing come possono controllare e gestire la qualità e la sicurezza dei rilasci del prodotto sviluppato prima che entri in produzione?

Diverse sono le problematiche sulla valutazione delle competenze di un fornitore di software. In generale, deve essere tenuto conto che lo sviluppo software è un'attività artigianale che si espone all'errore.
Inutile fidarsi ciecamente delle certificazioni aziendali, che attestano una "competenza aziendale" e non attestano le competenze dei singoli. In caso di outsourcing dello sviluppo dovrebbero essere valutate le competenze dei programmatori appurando che conoscano le metodologie di sviluppo sicuro e, nei casi più critici, dovrebbero essere eseguiti audit di terza parte sul codice mirati ad individuarne le criticità. Per avere dei risultati significativi sarebbe sufficiente fare analizzare ad una terza parte esperta circa il 20% delle righe di codice - pratica nota come code review, che può vagliare, quando fatta manualmente circa 300-500 righe di codice al giorno. L'audit, eseguito da una terza parte autorevole è un ottimo deterrente che innalza la qualità e, a conti fatti, conviene!

HSPI SpA - Consulenti di Direzione
Via Guido Reni 2/2 - 40125 Bologna (Italy)
P.IVA 02355801206

Tel. +39 051 2960545 | Fax. +39 051 6568695
BOLOGNA | MILANO | ROMA
www.hspi.it | e-mail: info@hspi.it