CENTRUL NAȚIONAL DE RĂSPUNS LA INCIDENTE DE SECURITATE CIBERNETICĂ

CERT-RO

GHID

SECURITATEA IN CICLUL DE DEZVOLTARE AL UNUI PRODUS SOFTWARE

Versiunea 1.0 – 26 octombrie 2012

Ghid dezvoltat cu sprijinul:

safetech innovations

www.safetech.ro

Cuprins

Introducere

Trendul ultimilor ani arată o evoluţie ascendentă îngrijorătoare a activităţilor de hacking îndreptate împotriva companiilor şi a resurselor acestora, în scopul determinării unor pierderi de imagine sau financiare importante şi a încetinirii activităţii acestora.

Grupurile şi curentele de hacktivism ridică din ce în ce mai multe probleme atât organizaţiilor care şi-au standardizat cea mai mare parte din activităţi la nivel informaţional, cât şi celor care nu folosesc practice solide de securitate în toate stadiile de dezvoltare a aplicaţiilor lor.
Ameninţările de securitate au căpătat astfel o importanţă din ce în ce mai mare. În acelaşi timp, a crescut interesul specialiştilor în domeniu pentru conştientizarea necesităţii dezvoltării unor metode noi de protejare împotriva ameninţărilor şi a vulnerabilităţilor, precum şi determinarea lor spre obţinerea unor standardizări de securitate în toate fazele evolutive ale aspectelor din organizaţiile care sunt vizate de hackeri.

Strategiile de securitate a informaţiilor pun accent din ce în ce mai mult pe integrarea componentelor, a activităţilor şi a soluţiilor de securitate în fiecare etapă de viaţă a activităţilor de business ale organizaţiilor, până la scalarea acestora în sisteme şi aplicaţii, acordând o mai mare atenţie tuturor acestor faze evolutive de dezvoltare a unei aplicaţii în mod individual.

SDLC

Într-o accepţiune unitară, **ciclul de viaţă al dezvoltării unui software (din engleză: Software Development LifeCycle – SDLC)** se referă la paşii structurali şi metodologia folosite pentru dezvoltarea unui produs de tip software. Termenul de Software Development LifeCycle este în strânsă legătură cu Systems Development LifeCycle, de cele mai multe ori ultimul înglobându-l pe primul la nivel conceptual.

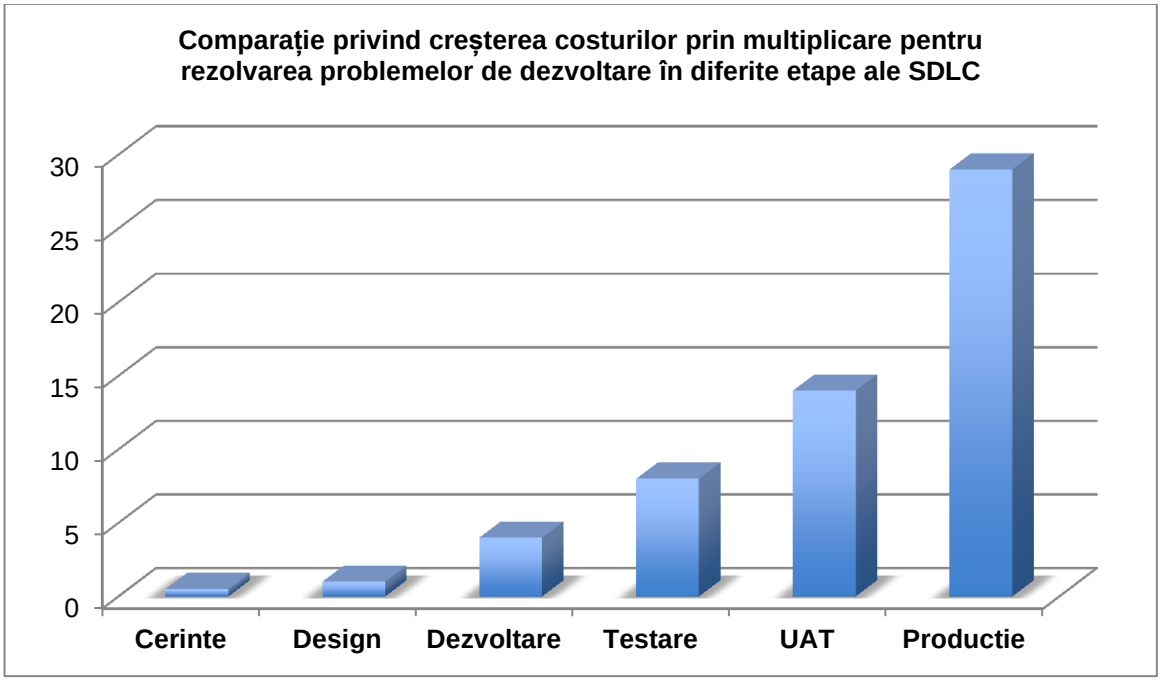
Datorită complexităţii tehnologice în creştere a sistemelor şi proceselor implicate în SDLC, creşte deopotrivă şi complexitatea vulnerabilităţilor disponibile la diverse niveluri ale unei aplicaţii, care dacă sunt exploatate, pot determina scăderea controalelor de securitate şi la nivelul altor niveluri ale aplicaţiei.
De exemplu, un port deschis în reţea poate determina descoperirea şi exploatarea unor sisteme neactualizate sau a unor vulnerabilităţi existente în anumite aplicaţii.

Pentru a asigura eficienta implementării unor controale de securitate, este necesară o abordare holistică asupra securităţii şi o viziune de ansamblu asupra tuturor componentelor, atât din perspectiva nivelurilor structurale fizice sau logice, cât şi la nivel de procese constructive, prin implicarea principiilor de securitate a informaţiilor în toate fazele SDLC.

Principalele motive pentru abordarea unei strategii de securitate bazate pe includerea securităţii informaţiilor în SDLC au ca premise necesităţi punctuale sau generice care reies din însuşi procesul de business.

Securitatea în SDLC poate fi interpretată ca un factor de succes pentru business, mai degrabă decât o barieră în cadrul realizării proiectelor, în special pentru că:

- activitatea de securitate este mult mai ieftină dacă este planificată încă de la începutul proiectului
- implementarea unor controale de securitate în sisteme la începutul proiectului în loc să fie implementate la sfârşit, poate creşte performanta sistemelor per ansamblu
- se poate reduce necesitatea costisitoare de reevaluare în post-producţie a unor controale şi măsuri necesare pentru scăderea riscurilor la un nivel acceptabil.



Având în vedere aceste aspecte referitoare la costurile care pot creşte exponenţial pentru rezolvarea problemelor de dezvoltare neidentificate sau neadresate la momentul descoperirii lor, este necesară incorporarea conceptelor de securitate în **fazele SDLC**: cerinţe, design, dezvoltare, punere în producţie şi eliminare.

Astfel se poate construi un **profil de securitate al software-ului** respective, bazat pe respectarea conceptelor de securitate esenţiale, generale şi de design, aşa cum sunt clasificate acestea mai jos:

Profil de securitate al software-ului	
Concepte esentiale	- confidențialitate - integritate - disponibilitate
Concepte generale	- autentificare - autorizare - audit si jurnalizare - managementul sesiunilor - managementul erorilor și excepțiilor - managementul parametrilor de configurare
Concepte de design	- cel mai mic privilegiu pentru a face ceva - separarea drepturilor - securitate pe niveluri - fail secure - mediere completă - open design - mecanisme de least common - acceptanța psihologică - legături intre componentele existente - celâa mai slabă verigă - un singur punct de avarie

Managementul riscurilor este unul dintre aspectele cheie ale managementului securităţii şi se înscrie în conceptele de baza de securitate cu care se poate lucra în SDLC.

Managementul riscurilor constă în evaluări preliminare asupra necesităţii controalelor de securitate, a identificării, dezvoltării, testării, implementării şi verificării controalelor de securitate, astfel încât niciun proces iniţiat în scopuri distructive se va încadra într-un nivel acceptabil de risc.

În sistemul de management al securităţii se înscriu şi aspectele legate de guvernanţă de securitate: politici, norme, standarde şi proceduri de securitate.

Etapele SDLC din perspectiva securităţii

Există mai multe accepţiuni ale modului de împărţire a etapelor SDLC din perspectiva integrării securităţii, iar acestea depind de complexitatea şi necesitatea urmărite în cadrul proiectului de dezvoltare a software-ului, însă majoritatea au în comun clasificarea generică de mai jos:

1. cerinţe pentru software securizat
2. design de software securizat

3. dezvoltarea de software securizat
4. testare de software securizat
5. producţie
6. mentenanţă şi eliminare

Cerinţe pentru software securizat

În cadrul etapei de **cerinţe pentru un software securizat** este foarte important să se documenteze cât mai exact cerinţele pentru software-ul care urmează a fi dezvoltat sau achiziţionat.

Lipsa acestor cerinţe iniţiale de securitate poate ameninţa respectarea nivelurilor de confidenţialitate, integritate şi disponibilitate. De asemenea, lipsa cerinţelor determina dezvoltarea unui software slab din punct de vedere calitativ, planificare depăşită de timp alocat resurselor din proiect şi cel mai probabil va determina creşteri cu costurile necesare reluării documentarii, analizei şi concretizării aplicării cerinţelor şi rezolvarea erorilor.

Cerinţele de securitate pot fi preluate din mai multe surse de documentare:

- politici, norme şi standard interne ale organizaţiei
- cerinţe de conformitate
- reglementări legale externe
- standarde internaţional

Clasificarea informaţiilor poate fi o bază pentru construirea cerinţelor de securitate iniţiale, întrucât în baza acestora se pot construi controale de securitate asignate datelor şi informaţiilor în funcţie de nivelurile de clasificare corespunzătoare.

Design de software securizat

Principiile legate de **designul de software securizat** pornesc de la premisa că toate cerinţele de securitate emise vor implementate în designul software-ului.

În această etapă se folosesc evaluări iniţiale asupra probabilităţii atacurilor, plecând de la:

- modelele de ameninţări (Threat Models).
- identificarea controalelor
- prioritizare în funcţie de riscul asupra afacerii

Abordarea şi urmărirea implementării cerinţelor de securitate şi alinierii designului cu politicile interne trebuie să se facă pe de-o parte concomitent cu procesul de design propriu-zis, apoi printr-

o reevaluare finală a designului înainte de a merge în faza de dezvoltare, astfel încât să se asigure faptul că triada CIA este în continuare acoperită în această fază.

Toate principiile de design trebuie să ţină cont de cerinţele de securitate şi de constrângerile mediilor în care va rula noul software. Astfel, cerinţele de securitate pot fi ajustate astfel încât să acomodeze noile cerinţe de mediu tehnologic, însă analiza finală a arhitecturii trebuie să se facă pe fiecare nivel în parte, pentru a se asigura faptul că toate controalele de tip defense în depth vor fi implementate corespunzător.

Dezvoltare software securizat

Un rol important în realizarea unui produs software securizat îl are programatorul. Pe lângă a asigura funcţionalităţile solicitate de business, codul dezvoltat trebuie să asigure şi controalele de securitate necesare pentru a putea proteja informaţiile procesate. Deşi rolul programatorului este de a rezolva probleme de business, produsul creat de el poate deveni o problemă pentru business dacă este creat fără a înţelege mecanismele de securitate la nivelul codului.

Este foarte importantă familiarizarea programatorilor cu vulnerabilităţi comune ale codului care se regăsesc în produsele software, dar şi înţelegerea modului în care un atacator va încerca să exploateze produsul software.

Conform OWASP TOP 10, cele mai importante vulnerabilităţi în cadrul aplicațiilor software sunt:

- Injection
- Cross-Site Scripting (XSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross-Site Request Forgery (CSRF)
- Security Misconfiguration
- Insecure Cryptographic Storage
- Failure to Restrict URL Access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

Pentru eliminarea vulnerabilităţilor de mai sus este necesar implementarea unor mecanisme de securitate, cum ar fi: validare input, protecţii criptografice, managementul memoriei, tratarea excepţiilor.

După elaborarea codului, acesta trebuie să fie analizat de către experţii în securitate pentru identificarea vulnerabilităţilor existente la nivelul acestuia.

Codul poate fi analizat:

- *Static*: inspecţia manuală sau cu ajutorul uneltelor automate a codului sursa, fără a fi executat;
- *Dinamic*: inspecţia codului la momentul execuţiei;

Testare securitate software

Pentru a putea fi considerat sigur, un produs software trebuie să fie testat din punct de vedere al securităţii. Testele de securitate vor confirma implementare corectă a mencanismelor de securitate şi faptul că acestea rezista la atacurile informatice.

Testarea securităţii este un întreg proces în ciclul de dezvoltare al unui produs software securizat. Rezultatele testaţii securităţii sunt foarte importante pentru evaluarea finală a calităţii produslui software. Orice produs software care a fost validat din punctul de vedere al securităţii are o calitate mai bună decât un produs software care nu a fost validat.

Testarea securităţii poate fi utilizată pentru a determina metodele şi oportunităţile prin care un produs software poate fi atacat şi se realizează de către experţi specializaţi în acest domeniu (“ethical hacker”).

Există mai multe metode de testare a securităţii unui produs software, fiecare din acestea având rolul de a identifica diverse categorii de vulnerabilităţi:

- *Whitebox*: testare de securitate având cunoştinţe despre arhitectură şi modul de implementare. Presupune acces la codul sursă şi la sistemul care găzduieşte produsul software. Prin această metodă se identifica cauzele exacte ale unei vulnerabilităţi;
- *Blackbox*: testare de securitate fără a avea cunoştinţe despre arhitectură şi modul de implementare a produsului software. Are rolul de a identifica vulnerabilităţile exploatabile din cadrul produsului;
- *Fuzzing*: testare de tip “brute-force” prin injectarea de date randomizate în vederea generării de erori la nivelul produsului software;
- *Scanning*: testare prin scanare automată de vulnerabilităţi;
- *Penetration testing:* simularea de atacuri reale ale hack-erilor.

Acceptanța de securitate a unui produs software

Înainte ca un produs software să fie pus în funcţiune acesta, împreună cu mediul pe care a fost instalat, trebuie să fie verificat de către experţii în securitate. Trebuie validat faptul că au fost implementate toate cerinţele de securitate, iar controalele implementate rezista în cazul unor atacuri informatice.

Un produs software nu trebuie pus în funcţiune până când nu a fost certificat şi acreditat că riscul residual se regăseşte la un nivelul acceptabil. În cazul în care există riscuri peste nivelul acceptabil, punerea în funcţiune trebuie aprobată şi asumată de către beneficiarul produsului software.

Beneficiile unei acceptante formale a unui produs software include validarea cerinţelor de securitate, verificarea controalelor de securitate şi asigurarea faptului că acesta nu este doar rezistent la atacurile informatice, ci şi în conformitate cu reglementările aplicabile.

Securitatea în timpul funcţionării

Din cauza evoluţiei tehnologice, noi vulnerabilităţi şi ameninţări sunt descoperite în fiecare zi. Pentru a menţine nivelul dorit de securitate al informaţiilor valoroase, aplicaţia şi infrastructurile adiacente trebuie să fie monitorizate în mod continuu, din punctul de vedere al securităţii. Procesul de management al vulnerabilităţilor este cel mai important proces pentru a menţine un nivel de securitate adecvat.

Monitorizarea de securitate se realizează prin:

- Scanări periodice automate de vulnerabilităţi;
- Teste de penetrare anuale sau chiar mai des în funcţie de nivelul de securitate dorit;
- Verificări periodice de conformitate cu standardele de securitate;
- Analiza jurnalelor de activitate
- Detecţia incidentelor de securitate.

Concluzie

Securitatea informaţiei are un rol foarte important în fiecare etapă din ciclul de dezvoltare a unui produs software. Lipsa implicării specialiştilor în securitate sau utilizării principiilor de securitate în fiecare etapă de dezvoltare poate duce la apariţia de vulnerabilităţi în cadrul produsului şi chiar anularea eforturilor depuse în celelalte etape de dezvoltare.