IMPLEMENTAREA SECURITĂȚII CIBERNETICE ÎN DEZVOLTAREA SOFTWARE

Maxim CATANOI

Departamentul Ingineria Software și Automatică, Facultatea Calculatoare, Informatică și Microelectronică, Universitatea Tehnică a Moldovei, Chișinău, Moldova

Autorul corespondent: Maxim CATANOI, e-mail: *maxim.catanoi@isa.utm.md*

Coordonator: **Dumitru CIORBĂ**, dr., Universitatea Tehnică a Moldovei

Rezumat. *Creșterea numărului de amenințări cibernetice induce considerarea securității încă de la începutul ciclului de dezvoltare software. Experiența companiilor dezvoltatoare de software arată că numărul de vulnerabilități ce pot fi identificate la etapa finală de dezvoltare se pot diminua la o abordare corectă și sistematică a aspectelor de securitate cibernetică. Lipsa unei protecții cuvenite nu se observă în urma testelor funcționale a sistemelor software, ceea ce deseori rezultă în compromiterea datelor ce sunt procesate sau stocate de către aceste sisteme. În lucrare se analizează necesitatea elaborării unui cadru ce ar reieși din experiențe practice fiind urmată pentru a asigura un nivel de protecție avansat încă de la primele faze de dezvoltare. Cadrul fiind versatil ar cuprinde atât echipele de dezvoltare software, cât și clienții ce solicită sisteme informatice complexe. O atenție sporită se acordă sistemelor software care sunt accesibile din rețeaua globală Internet, acestea fiind expuse atacurilor cibernetice în mod continuu, evidențiind că doar abordarea corectă și metodologică pe tot parcursul de dezvoltare software poate diminua efectul compromiterii datelor, precum și a infrastructurilor conexe, în urma exploatării de vulnerabilități rezultate a unor omisiuni de securitate.*

Cuvinte cheie: *dezvoltarea software, securitate cibernetică, amenințări cibernetice, cadre de dezvoltare, threat modeling, secure coding, security testing, system hardening, SSDLC.*

Introducere

Abordarea curentă de software pune accentul, în mare parte, pe dezvoltarea cu resurse minime și în termeni cât mai restrânși, iar aspectul securității cibernetice, de regulă, rămâne neacoperit. Aceasta cauzează apariția de vulnerabilități care expun unui risc considerabil sistemele informaționale, inclusiv datele ce sunt stocate sau procesate de către acesta. Astfel în ultimii ani se atestă o creștere a vulnerabilităților în sistemele software (Figura 1) [1]. Ipoteza fiind că majoritatea din acestea se datorează lipsei unui cadru de dezvoltare software care ar presupune abordarea aspectului de securitate cibernetică încă de la primele etape de dezvoltare.

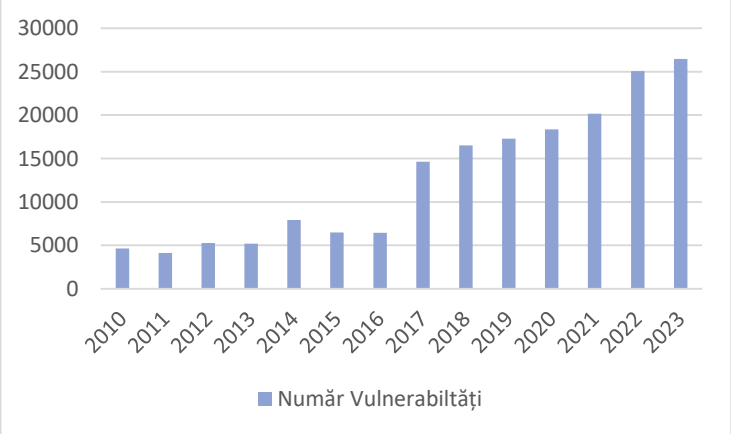

Figura 1. Trendul vulnerabilităților în ultimii ani [1]

Un alt studiu indică faptul că nerespectarea unui cadru de dezvoltare software în mod securizat, ar crește considerabil eforturile de corectare a vulnerabilităților identificate la ultima etapa de dezvoltare, care ar reprezenta costuri de 30 de ori mai mare comparativ cu celea care ar putea fi alocate în cazul în care aspectul de securitate cibernetică ar fi abordat corect încă de la etapele incipiente (Figura 2).

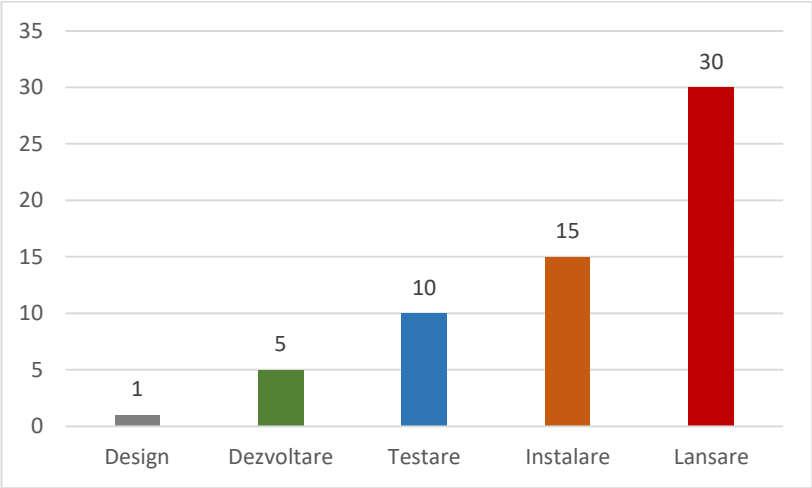

Figura 2. Costuri de referință de eliminare a vulnerabilităților [2]

Abordarea corectă a aspectelor de securitate cibernetice ar satisface cerințele reglementărilor și standardelor din domeniu în special atunci când ne referim la sisteme informaționale ce stochează, procesează sau transmit date ale cardurilor bancare, care s-ar supune reglementărilor PCI-DSS [3].

Problema de cercetare

În spațiul cibernetic orice linie de cod neasumată ar putea reprezenta o vulnerabilitate care ar aduce la consecințe drastice în condițiile în care aceasta ar fi exploatată de către actorii rău intenționați. Aceste riscuri sunt în creștere datorită nivelului de complexitate crescută a sistemelor. Din aceste considerente este necesară redefinirea proceselor de dezvoltare software ce ar stipula cum trebuie abordat aspectul de securitate cibernetică. Totodată se propune schimbarea accentelor de la o abordare reactivă în cazul confruntării cu consecințele unui atac cibernetic, spre un model pro-activ care integrează măsuri de securitate robuste încă de la începutul ciclului de dezvoltare software. Astfel s-ar putea de asigurat ca sistemele software nu sunt doar funcționale și performante, dar și rezistente în fața amenințărilor curente.

Metodologia cercetării

În urma studiului dat s-a propus respectarea unei metodologii de cercetare ce ar evidenția cadrele existente aplicabile prin analize bazate pe experiența proprie. Analiza celor mai populare cadre existente transpuse prin prisma realităților curente de dezvoltare software ar accentua necesitatea ajustării cadrelor existente sau elaborarea unui cadru nou ce ar fi aplicabil în diverse situații. Au fost analizare cadrele de referință de securitate cibernetică utilizate în industrie. Analiza dată a permis identificarea avantajelor și dezavantajelor acestora.

Cadrele existente

Pentru analiza dată au fost selectate trei cadre existente care sunt utilizate de către cele mai mari companii cu renume, și anume:

1. Microsoft Security Development Lifecycle (SDLC) [4]
    a) Avantaje: Dezvoltat de o companie majoră cu experiență vastă.

b) Dezavantaje: Complexitatea ridicată ce poate fi un impediment pentru organizații mici.

2. OWASP Software Assurance Maturity Model (SAMM) [5]
   a) Avantaje: Este un cadru echilibrat în raport cu celelalte metode.
   b) Dezavantaje: Necesită mai mult timp pentru implementări.
3. NIST SP 800-64 [6]
   a) Avantaje: Utilizează standarde și principii recunoscute internațional, cu o abordare testată în timp.
   b) Dezavantaje: Complexitatea poate împiedica implementarea în organizațiile mici care nu sunt familiare cu standardele NIST.

Concluzii

Implementarea cadrelor de referință necesită o investiție semnificativă de timp și resurse, ceea ce poate fi dificil pentru organizațiile mici și mijlocii. Prin urmare se atestă o necesitate pentru elaborarea unui cadru de securitate pentru procesul de dezvoltare software. Caracteristicele de bază a cadrului nou propus spre elaborare se vor baza pe următoarele principii:

1. Simplificare – reducerea complexității de implementare, concentrând-se pe elementele cele mai critice ce ar oferi un beneficiu mai mare în raport cu efortul investit
2. Flexibilitate – capacitatea de adaptare la schimbările rapide în domeniul tehnologiilor informaționale și la specificul diverselor organizații ce ar permite ajustări rapide și eficiente
3. Practicabilitate – procese clare și concise, cu exemple de tehnici care ar facilita implementarea rapidă pe diferite scenarii de dezvoltare
4. Resurse adaptate – oferirea de instrumente și resurse specifice pentru organizații cu bugete și echipe restrânse, facilitând astfel adoptarea și menținerea securității.

Aceste principii ar putea permite elaborarea cadrului care ar fi mai simplu, mai ușor de aplicat la livrarea rapidă a unor sisteme informatice prototip (MVP, Minimum Viable Product), care deseori sunt lipsite de protecție elementară împotriva atacurilor cibernetice.

Referințe:

[1] Qualys, 2023 Threat Landscape Year in Review: If Everything Is Critical, Nothing Is, Ianuarie 2024, [Online]. Available: https://blog.qualys.com/vulnerabilities-threat-research/2023/12/19/2023-threat-landscape-year-in-review-part-one
[2] NIST, 2002, The Economic Impacts of Inadequate Infrastructure for Software Testing, Mai 2002, [Online]. Available: https://www.nist.gov/system/files/documents/director/planning/report02-3.pdf
[3] PCI-DSS, 2022, PCI DSS v4.0 At a Glance, Decembrie 2022, [Online]. Available: https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Supporting%20Document/PCI-DSS-v4-0-At-A-Glance.pdf
[4] Microsoft, 2011, Simplified Implementation of the Microsoft SDL, Februarie 2011, [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=12379
[5] OWASP, 2022, Software Assurance Maturity Model, Ianuarie 2022, [Online]. Available: https://github.com/OWASP/samm/raw/master/Supporting%20Resources/v1.5/Final/SAMM_How_To_V1-5_FINAL.pdf
[6] NIST, 2019, NIST Special Publication 800-64 Revision 2, Mai 2019, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-64r2.pdf